Musicons: Visualisation of music to improve user-guided search and exploration

Version of May 13, 2023

Vera Hoveling

Musicons: Visualisation of music to improve user-guided search and exploration

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Vera Hoveling
born in Groningen, the Netherlands

Computer Graphics and Visualization Group
Department of Intelligent Systems
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Cover picture: icons for classical music in a playlist setting

Musicons: Visualisation of music to improve user-guided search and exploration

Author: Vera Hoveling
Student id: 4591941

Abstract

This document describes the work performed the master thesis on a new approach to visualisation for music. We investigate how visualisation of (latent) characteristics of a song can improve user-guided search and exploration. In this document, the need for such visualisation is described, followed by a formulation of the research questions, literature reviews, our method, design and evaluation, a discussion of our results and an outlook on further development of this project.

Thesis Committee:

Chair:	Prof. Dr. E. Eisemann, Faculty EEMCS, TU Delft
University supervisor:	X. Luo, Faculty EEMCS, TU Delft
Committee Member:	Prof. Dr. J.A. Martinez Castaneda, Faculty EEMCS, TU Delft

Preface

This project could not have been brought to completion without the help of many many people. I want to thank a few of them.

First and foremost, Professor Elmar Eisemann, for convincing me to enroll in the master's programme, for the supervision of not only my master thesis, but my bachelor's and honours as well and for granting me an extraordinary amount of freedom in each of those projects. I am grateful to Xuejiao Luo who was always ready to answer my questions, provided insightful comments and valuable feedback, which greatly enhanced the quality of this work. I want to acknowledge Professor Jörn Loviscach, for the stimulating correspondence and digging up 17 year old studies. Then some invaluable TU Delft staff: Michel Rodrigues, who coached me through post-covid recovery and Lauretta Ritchie, who scheduled countless appointments and always remained very kind when I pushed for an earlier or longer meeting.

I also want to thank Philipp Darkow, for endlessly testing my user tests, Sven Rohde, for showing me the way in server-side country, Priscilla Haring-Kuipers, for scrutinising my user study methods, Stijn Kuipers-Haring, for being inspiring for almost a decade now and Sayra Gmelig Meyling, world's most overqualified rubber duck.

I am deeply grateful to Jacques Povee for his unwavering presence and support, to Mom & Dad, for everything, and my friends Beryl, Carolijn and Loes, for checking in regularly, listening, and thoughtful discussions during challenging times.

And of course, I want to acknowledge my fellow students, who have made my master's worthwile: Marijn Roelvink, Eva Slingerland, Sam Vijlbrief, Julian Biesheuvel, Niels Hoogerwerf, Yorick de Vries, Maaike Visser, Julian van Dijk and many more not mentioned here.

Vera Hoveling
Amsterdam, the Netherlands
May 13, 2023

Contents

List of figures

Chapter 1

Introduction

This chapter introduces the motivation for this thesis and formulates the research questions.

1.1 Motivation

Music streaming services are increasingly becoming music discovery services [Chodos et al., 2019; Hosey et al., 2019]. Users discover new music either through algorithmic recommendations, or via user-guided search and exploration. The latter has shown to result in more diverse listening, which is considered a desirable metric in the music industry as it is related to long-term user engagement [Anderson et al., 2020].

Despite the progress made in enhancing users' navigation of music collections through algorithmically enhanced search, most interfaces have yet to introduce innovative visual cues. Icons for songs are - still - constrained to album art. For a user of music streaming services, the only way to find out what a song is like, is to listen to it.

For example: Spotify creates personalised playlists for users, such as the 'Discover weekly' playlist that can be seen in Figure 1.1. Assuming this is indeed unseen music from unknown artists, waiting to be discovered, the user cannot infer from artist names or album art what kind of music it concerns, until listening.

Listening a song to identify what music it is, or if it matches some expectation, takes a couple of seconds, at the very least. The hypothesis is that visualisation of characteristics of a song (e.g. mode, tempo or mood) can improve user-guided search and exploration by enabling visual search [Wolfe, 2015]. We further hypothesise that such visual search may speed up search times when searching with an open or exploratory mindset [Hosey et al., 2019]. Earlier work has demonstrated that visual identifiers speed up UI navigation [Lewis et al., 2004].

1.2 Research questions

We formulate the main research question as follows:
How can a visual representation help speed up identification of a song with distinctive characteristics?

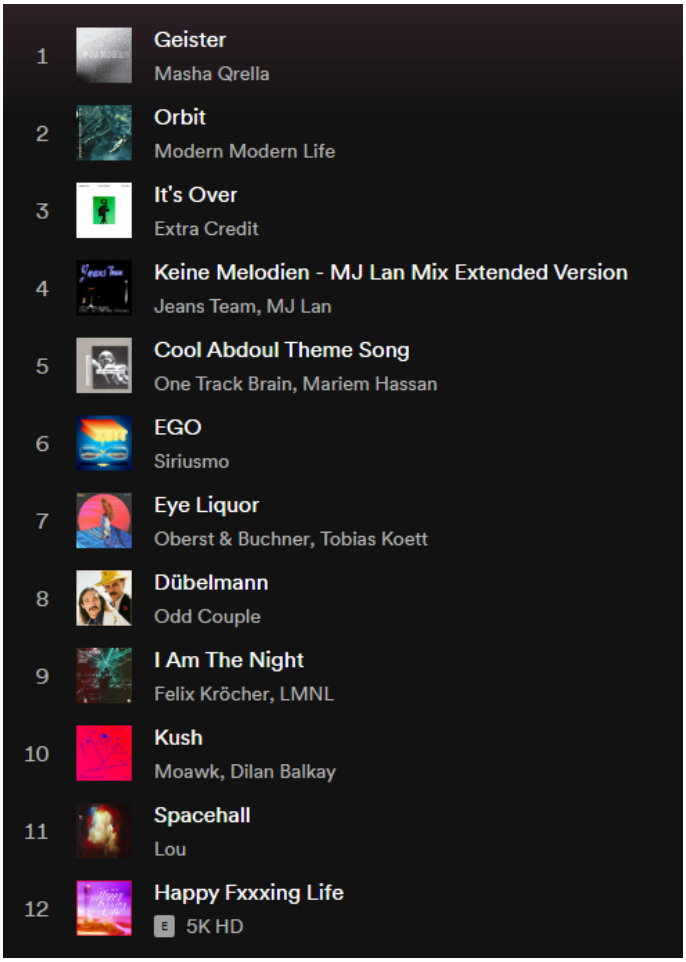


Figure 1.1: Spotify's automatically generated playlist 'Discover Weekly' shows little (visual) information that relates to the music content.

To answer the main research question, we address the following sub-questions:

1. What are distinctive characteristics of a song and which features can be used to represent those?
2. What role can latent variables play in the representation of characteristics of a song?
3. How can features be mapped to a visual representation?

Chapter 2

Related work

This chapter discusses technologies and academic literature related to music features, latent representations for music, aspects of visualisation, custom icons and glyphs. It is structured in four sections, which each explore the literature that relates to a sub-question.

2.1 Music Characteristics and Features

In this section, we review related work on music characteristics and feature extraction. This section addresses in particular the first sub-question: **What are distinctive characteristics of a song and which features can be used to represent those?**

To find an answer to this question, this literature review aims to provide a very high-level overview of music characteristics and features that model those, as well as features that more implicitly capture characteristics. By providing this context, we narrow down the position of this work and clarify on what aspects of music we base our design.

2.1.1 Distinctive characteristics of music

The first question in our sub-question: *What are distinctive characteristics of a song?* may invite a comprehensive study of music theory. Taking into account our main research question, and keeping the scope of our answer contained, we will constrain ourselves to a computer science perspective.

Perception of music and its characteristics are determined by properties of both the music and whoever is listening to it [Schedl et al., 2013]. Although we do incorporate the user in our evaluation, we focus our research on the music properties. In modelling music features, we distinguish content- and context-based features. Content-based features are derived from the music signal only, whereas context-based features are derived from additional information, such as tags, lyrics, reviews and other cultural information. We will focus on content-based features, which is often the only information available for new unseen music [Chodos et al., 2019]. Since content-based features fall within the domain of Music Information Retrieval (MIR), our focus will primarily be on this area of research.

2.1.2 Content-based Features

In music content's extracted features, low-level, mid-level, and high-level features can be discerned. Low-level features, such as a statistical summary of the signal, closely resemble the original representation. Mid-level features are typically a combination or extension of low-level features that integrate additional knowledge. High-level features are concepts as humans understand it and they carry semantic meaning. The boundaries between the levels are not strict but rather blurred [Knees and Schedl, 2016].

In MIR, the dominant approach to feature extraction used to be problem-specific modelling, based on statistical analysis and algorithmic processing of the audio signal. Features were explicitly modelled with extensive domain knowledge, a practice now somewhat derisively referred to as 'hand-crafted features'. In recent years, however, the approach has been more deep learning oriented. Deep learning models are used for modelling high-level features but also for other MIR tasks [Müller et al., 2022]. Deep learning models for MIR rely much less on feature modelling than previous approaches: they either learn in an end-to-end fashion or from low-level features.

2.1.3 High-level features

High-level features are the most semantically meaningful and are also the concepts with which end-users reason about music. Amongst high-level features, there is a hierarchy too: from more atomic high-level features such as tempo, mode and key, more complex semantic features can be determined, such as instrumentation, genre and emotion/mood. These types of high-level features are very interesting, yet at the same time not well defined: there is only 80% agreement amongst humans on genre classification [Schedl et al., 2014] and there exist varying cultural interpretations on more complex emotions in music [Lee et al., 2021].

Up until recently, it was the standard approach to construct high-level features from mid- and low-level features, by either algorithmic rules or machine learning algorithms [Müller et al., 2022]. The dominant approach is turning towards end-to-end deep learning models that extract features from a raw signal or from very low-level features.

2.1.4 Popular features in research

High-level features that are popular in research are available from the Million Song Dataset (MSD) [Bertin-Mahieux et al., 2011] and the Spotify Web API. The MSD is a widely used benchmark in MIR, providing both content and context based features. The MSD considers itself a representative dataset of recent western commercial music [Bertin-Mahieux et al., 2011].

The features of the MSD and the Spotify Web API overlap. This stems from their shared origin in The Echo Nest, a music intelligence 'platform'. The documentation of MSD notes that their features contain "almost all the information available through The Echo Nest API for one million popular tracks" [Bertin-Mahieux et al., 2011]. Spotify acquired The Echo Nest in 2014. In 2016, The Echo Nest's API was merged into the Spotify Web API [Skidén, 2016]. We still observe that many of the feature descriptions in the Spotify Web API are literally the same as they were in The Echo Nest's API.

A popular investigation in literature is the use of Spotify features to predict the success of new products [Sciandra and Spera, 2022; Gulmatico et al., 2022; Nijkamp, 2018]. This is an objective made rather easy by Spotify, as the Spotify Web API also presents a popularity rating for each track. We observe that the literature focuses mainly on the very high-level features that are available from the Spotify Web API, via the 'Audio Features' API call, rather than the slightly lower-level 'Audio Analysis' API call.

Although we acknowledge that the Spotify Web API features are of high quality and popular in research, we object against using private, closed source data. We have concerns for reproducibility, explainability and open research, amongst others. Therefor we have decided not to make use of the Spotify features directly, but we do use them in Section 3.3 as a benchmark to compare the performance of other features against.

2.2 Latent variables and representation learning in MIR

In this section, we review related work on latent-variable modelling for music. In particular, this review addresses the second sub-question: **What role can latent variables play in the representation of characteristics of a song?**

To find an answer to this question, we start with a definition of latent variable representations. We then investigate the use of latent variables in MIR and gloss over some popular deep learning techniques.

2.2.1 Learning representations and features

Learning latent representations is a machine learning approach to infer latent variables from observed events [Kopf and Claassen, 2021]. The goal of learning latent variable representations is to obtain abstract and useful representations from raw data for tasks such as classification and prediction [Latif et al., 2020].

It seems that the preferred terminology has moved from 'feature vector' to 'representation': at ISMIR 2021, 10 papers mentioned the word 'representation' in their title, as opposed to only 2 mentioning 'features'. In this document, we often omit the term 'latent' and abbreviate to 'representation learning', as in much of the literature on this topic, but note that by this we refer to a latent-variable representation.

2.2.2 Representation learning in MIR

Although many music-related tasks are well understood, often even intuitively, it is difficult to point out where relevant information is in the music data and how to describe this information in a numeric representation [Kim et al., 2020]. This difficulty shows in the failure of signal-processing based methods to represent all explanatory factors of variation behind music data [Bengio et al., 2013]. The emerging idea is that latent variables make sense for the representation of characteristics of a song, as they can learn more abstract features that better represent the audio [Hamel and Eck, 2010].

No surprise then, that in MIR representation learning is a very active research topic. Over 11.800 papers return from a Google Scholar search for "'representation learning'

AND music". Representation learning has, since decades, been investigated in MIR using machine learning models such as Hidden Markov Models and Support Vector Machines [Hamel and Eck, 2010]. The deep learning paradigm brought large advances with the possibilities of 'deep music representations' [Bengio et al., 2013] and has become the dominant approach [Müller et al., 2022].

A challenge of representation learning is the difficulty in establishing a clear objective or target for training [Bengio et al., 2013]. We observe in the literature that often representations are trained for a particular downstream task and then their representations inspected. In this sense, the latent representations are implicitly modelled [1] .

Downstream tasks in MIR representation learning are music generation [Dhariwal et al., 2020; Roberts et al., 2018; Pati and Lerch, 2021], transcription [Boulanger-Lewandowski et al., 2012], source separation [Chandna et al., 2017], various classification tasks [Han et al., 2016; Jeong and Lee, 2016; Choi et al., 2016; Lee et al., 2017; Choi et al., 2017] and recommendation [Van den Oord et al., 2013; Martín, 2017]. In addition, work is being done on finding a more generalised approach that is suitable for multiple tasks [Kim et al., 2020; Durand and Stoller, 2022].

Research has also been done on feature learning from content combined with metadata [Alonso-Jiménez et al., 2022; Park et al., 2017; Kim et al., 2018; Lee et al., 2019]. However, narrowing down our scope, we stick to a content-based approach, anticipating the availability of only an audio signal at run-time.

2.2.3 Techniques for learning latent representations in MIR

Representation learning methods in MIR draw upon methods from image processing and Natural Language Processing. What follows here is an outline of the three most prevalent deep learning architectures and methodologies, which have achieved the state-of-the-art in numerous MIR research domains [Müller et al., 2022]. For state-of-the-art models for each of those approaches, we refer to Table B.1.

Convolution Neural Network (CNN)

CNNs are considered well-suited for representations as their intermediate feature representations are shared and their hierarchical structure allows them to operate on multiple timescales [Van den Oord et al., 2013]. Do note that when using only frames of the data, the long-term dependencies of the music are lost [Martín, 2017]. It has been noted that CNNs are biased toward texture [Luo et al., 2021; Geirhos et al., 2018]. In spectrograms, the popular input for CNNs in MIR, texture is often interpreted as timbre of music [Won et al., 2022].

Variational Autoencoder (VAE)

The VAE [Kingma and Welling, 2013] proposes an improvement over the autoencoder (AE). Whereas the encoder of the AE outputs vectors in the latent space directly, the en-

[1] A notable exception is the seminal work of Van den Oord et al. [2013], who first modelled latent variables with Weighted Matrix Factorisation and then used those as learning targets.

coder of the VAE outputs parameters of a pre-defined, constrained, distribution in the latent space: the mean and standarddeviation of a Gaussians in each dimension. This constraint ensures the latent space is regularised. The benefit of a regularised latent space is that the whole latent space can be sampled and that latent variables are smoother. This is considered beneficial for generation as well as for representation learning [Bengio et al., 2013].

Transformers

The Transformer architecture [Vaswani et al., 2017], is a representation model relying on an attention mechanism to learn global dependencies between input and output. The attention mechanism facilitates the model to relate pairs of positions in a sequence learn long-term context by doing so. Transformers are a very popular approach to all kinds of sequential data. The many applications of Transformers in MIR are described in Won et al. [2022].

2.3 Icons and Visualisation for music discovery

In this section, we review the literature on music visualisation. Together with Section 2.4, this review addresses the third sub-question: **How can features be mapped to a visual representation?** We first discuss the idea of custom icons, then introduce our main inspiration Kolhoff et al. [2008] and subsequent works. We then discuss other work related to visualisation of music for discovery and retrieval. Finally, we introduce a framework that enables us to contextualise this work within the broader scope of visualisation research.

2.3.1 Custom icons

The concept of custom icons for files was popular for a couple of years after the publication of Lewis et al. [2004] and Setlur et al. [2005]. The idea of speeding up visual search with distinctive icons/glyphs originates from Lewis et al. [2004]. They propose 'VisualIDs': automatically generated visually distinctive icons, based on the hashed filename. Examples of the icons as generated by Lewis et al. [2004] can be seen in Figure 2.1. Setlur et al. [2005] offered a content-based approach with 'Semanticons', where the icons ought to reflect the gist of contents of the file. Although VisualIDs do not reflect the contents of files, their approach is to generate similar-looking icons for files with highly similar names, recognising the need for such a functionality.

Although some research has been done on the application of unique custom icons in specialised fields, such as software programming [Strobelt et al., 2009; Maguire et al., 2012; Stach et al., 2007], they have not gained widespread use in desktop applications, except for the 'Identicon'. Identicons serve as unique profile pictures for online applications, and are widely used in various online platforms, most notably Github.

Icons are frequently composed of glyphs, which are graphical representations of characters or symbols. In Section 2.4, we provide a comprehensive discussion of glyphs, which covers their exact definition and their place in visualisation research.

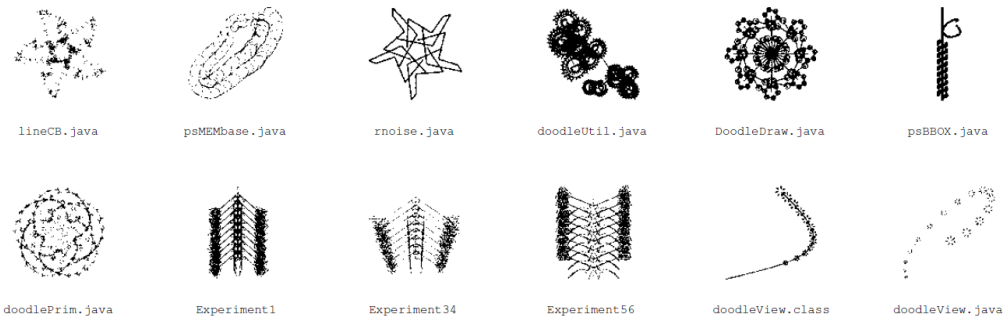


Figure 2.1: VisualIDs as generated by Lewis et al. [2004].

2.3.2 MusicIcons

Kolhoff et al. [2008] introduced the concept of a content-based icon for music, which they called 'MusicIcons'. They designed a parameterized 'flower-glyph' assigned to music files based on music features and user input, as shown in Figure 2.2. Their approach is strongly content-oriented and makes the visual clustering one of the main goals. It even allows for sorting based on the generated visualisation parameters.

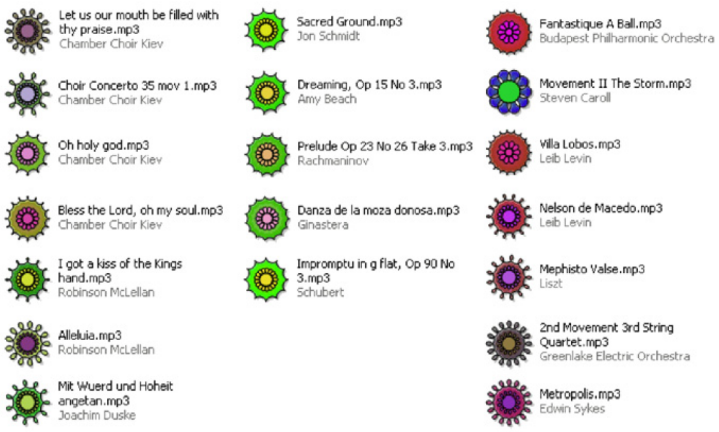


Figure 2.2: Flower glyphs as generated by Kolhoff et al. [2008].

The framework proposed by Kolhoff et al. [2008] is illustrated in Figure 2.3. As features, they used clustered MFCCs. To map features to glyph parameters, a neural network was trained on a small number of user-provided samples. The network learned a mapping from the clustered MFCCs to eight parameters. These parameters were used to generate the glyphs: two parameters regulate the number and shape of petals and the remaining six parameters are mapped to two sets of RGB colours, an inner colour and an outer colour.

To enable the user to explore music collections Kolhoff et al. [2008] also presents the user with a novel interface: a 2D plot with the glyphs of the songs, where the position of each glyph is determined by a 2D PCA projection of the feature vectors of the songs, as can

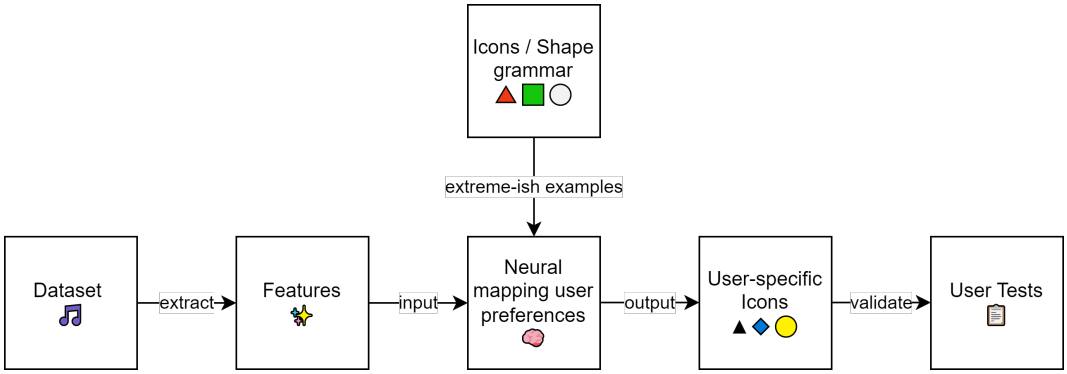


Figure 2.3: System as proposed by Kolhoff et al. [2008].

be seen in Figure 2.4.

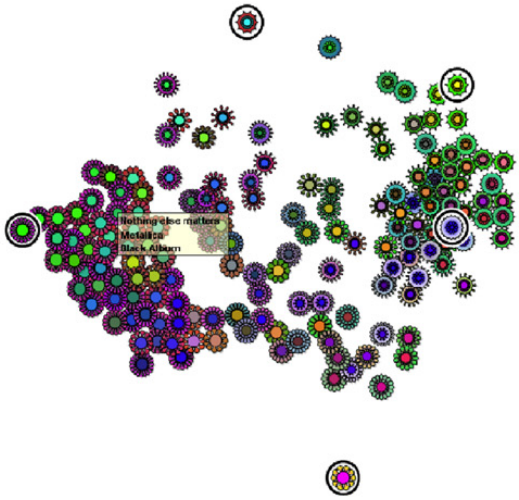


Figure 2.4: Plot of glyphs for music files, arranged according to the 2D PCA projection of the glyphs parameters. Image from Kolhoff et al. [2008].

We identify this work as our main inspiration and build upon their approach to colour. We propose significant changes to both the shape of the glyph and the feature selection process as well as the user interface. Our proposed adaptations are explained in detail in Chapter 3.

2.3.3 Work inspired by MusicIcons

The content- and music-oriented work by Kolhoff et al. [2008] seems to have sparked some projects in the years after its publication: Kusama and Itoh [2011]; Machida and Itoh [2011]; Kim et al. [2009]; Chen and Klüber [2010]; Uota and Itoh [2014]; Oda and Itoh [2007]. These works focused on trying other ways to browse music visually or generate music icons: tasks related to our research question. In addition, we have found work that does not

cite Kolhoff et al. [2008] as previous work but we consider it very closely related, such as Yoshii and Goto [2007] and Lima et al. [2019]. In this subsection, we will briefly highlight the three works we consider most relevant for our project.

Kim et al. [2009]

Kim et al. [2009] presents a system that 'generates icons that reflects the emotion of music for searching and finding music more efficiently'. The authors aim to use high-level features by designing parameterised shapes whose parameters are mapped to a 2D arousal-valence model [Thayer, 1990]. The icon for each song is then determined by the parameter settings associated with their respective location in the arousal-valence plane. Examples of images generated as by Kim et al. [2009] can be seen in Figure 2.5.

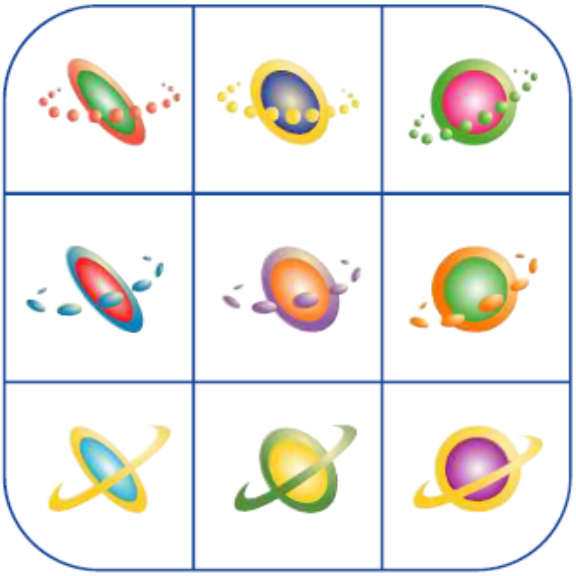


Figure 2.5: Affective music icons generated by Kim et al. [2009].

We find that their work is aesthetically pleasing and also employs a double colouring, which we too take from Kolhoff et al. [2008]. We also share an interest in generating the glyph with a parameterised shape. However, this shape is not very expressive and always offers a variation of a planet with rings.

Chen and Klüber [2010]

Chen and Klüber [2010] aims to facilitate music browsing and searching with automatically generated icons ('thumbnails'), much like our work. Their design, which can be inspected in Figure 2.6, relies on colour coding for genre and symbolic notation for all other information.

We find this work interesting as we share a similar objective with the authors of the paper. However, we find their method to be ineffective. Their approach employs textual

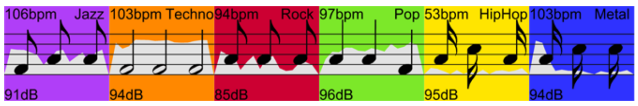
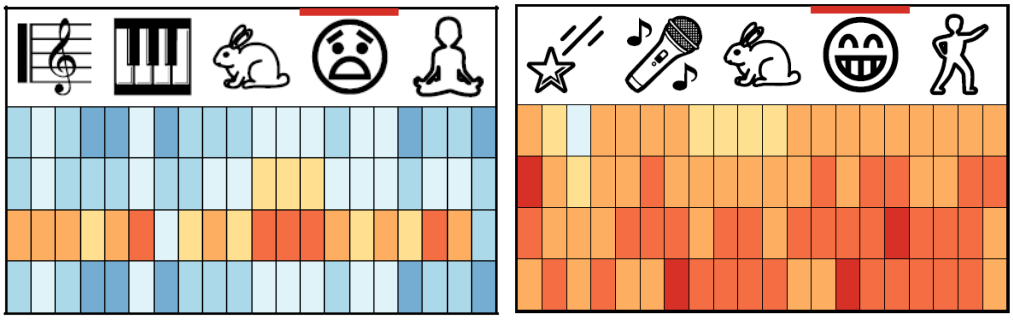


Figure 2.6: Music summaries as designed by Chen and Klüber [2010].

and staff notation, which leads to an icon that requires a significant amount of time to process. Moreover, their utilisation of binned categorical colours leaves little nuance for the intricacies and blurred classifications we find in music.

Lima et al. [2019]

Lima et al. [2019] does not mention Kolhoff et al. [2008] as previous work, but we do find that it functions very much in the same spirit of stimulating visual search for users, as they mention their objective to ”aid users on tasks related to exploration/browsing of music libraries and at queries for similar tracks based on visual characteristics”. To do so, Lima et al. [2019] extracts high-level features on mood, danceability, tempo, music genre and instrument, which are then mapped to a semantically meaningful visualisation. Their overall design can be inspected in Figure 2.7.



(a) Prelude In G Minor by Chopin (b) Mama Mia by Abba

Figure 2.7: Results of Lima et al. [2019] for Chopin and Abba, respectively.

Despite their ambitious intentions, we found the overall design to be somewhat lacking in effectiveness: the resulting image is made up of many components, of which some really take a while to process, like the rabbit or the falling star. It seems to us not the best way to facilitate visual search and suspect there must be better ways to do so.

2.3.4 Music visualisations for discovery and retrieval

Music visualisation techniques have been employed to facilitate search and discovery of music. Such techniques include the use of audio feature visualisations and interactive visual interfaces that allow users to explore music collections, like the 2D plot with song glyphs as presented by Kolhoff et al. [2008].

Visualisations for discovery and retrieval are often about how songs relate to each other as well as their place in a global structure [Knees et al., 2020]. There have been more works proposing some form of 2D embeddings of high-dimensional data for exploring and discovering music: Donaldson [2007]; Sprague et al. [2008]; Muelder et al. [2010]; Paulovich et al. [2011]. Self-organising maps were also popular to facilitate music discovery for a while [Khulusi et al., 2020]. Most of these works were published around the time Kolhoff et al. [2008] was published. Lately, the more effective t-SNE projection has been proposed [Lionello et al., 2018; Atassi; Anderson et al., 2020; Schedl, 2017; Benjamin and Altosaar, 2015; Shen et al., 2020]. However, such views have thus far never integrated in consumer-grade UIs. This may be due to the amount of resources that would be required for a large-scale t-SNE embedding.

Finally, also other novel ways for discovering music have been proposed, based more on the users input. For example, Knees and Andersen [2016] proposes to do retrieval based on sketches of timbre.

2.3.5 Formalising Visualisation Design

In this section we adopt a high-level approach to visualisation design and analysis. To position this work in a larger body of visualisation research, we use Tamara Munzners nested model [Munzner, 2009]. Munzners nested model provides a generic framework, widely applicable to visualisation research that enables us to reason and contextualise this work.

The model consists of four nested layers: domain, abstraction, idiom and algorithm. Their composition relative to each other can be seen in Figure 2.8. Each layer can be explained by the question(s) it poses:

- Domain: Who are the target users?
- Abstraction: What is shown? (data abstraction) & Why is a user looking at it? (task abstraction)
- Idiom: How is it shown? (visual encoding (how to draw) & interaction (how to manipulate))
- Algorithm: How to compute? (efficient computation)

We identify our work to be a problem driven work, which implies that our approach is to work from the outer layer to the inside layer. Although shown with the arrows as a one-way street, it is all too common to re-iterate many of the steps in the model [Munzner, 2014]. In addition, most research works address a few, and not all, of the layers in the model within one research carried out, some research just focuses on one layer exclusively. We further formalise our work in terms posed by the nested model in Section 3.2.

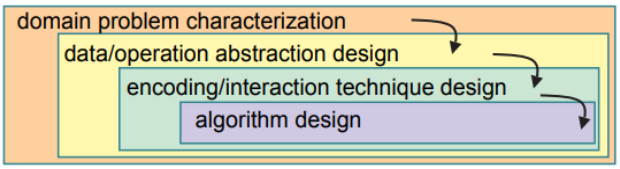


Figure 2.8: The four nested layers of Munzners visualisation model, from Munzner [2009].

2.4 Glyphs

In this section, we review the literature on one specific idiom in visualisation: glyphs. Together with Section 2.3, this review addresses the third sub-question: **How can features be mapped to a visual representation?**

We start with a definition of glyphs, then look at glyphs in visualisation research. We will then focus on star glyphs, delve into visual channels, specifically the use of colour and curvature, and explore data mapping. Finally, we will discuss the sorting of glyphs.

2.4.1 Definition

Although it is generally understood that glyphs are some graphical representation, the exact definition of a glyph seems somewhat diffuse [Chen et al., 2022] and unsurprisingly then, the term 'glyph' is used ambiguously in visualisation literature [Munzner, 2014]. In fact glyphs seem so hard to exactly define that Borgo et al. [2013] proposes two definitions. In a recent survey of visualisation literature, Chen et al. [2022] notes that the definition of glyphs is "notoriously hard" but that, most commonly, glyphs have been defined as "representations of multi-variate data".

Often, glyphs are composed of several geometric elements and visual channels, which encode different data channels. This allows for glyphs to show multiple attributes at once [Munzner, 2014] and makes them very suitable for high-dimensional data visualisation [Kammer et al., 2020; Keck et al., 2017] and tabular data [Brehmer et al., 2021]. The design space for glyphs is immense and far from fully explored [Borgo et al., 2013].

2.4.2 Glyphs in visualisation research

Glyphs have been an active topic of visualisation research since the 1970's. Some classic, well-known, well-established and well-researched glyphs are Chernoff faces, star glyphs and stick figures. A specific type of glyphs that has received considerable attention is the small versions of simple charts, such as bar charts [DuToit et al., 2012].

Recently more generalised frameworks to design semantically meaningful glyphs have been published [Khawatmi et al., 2022; Ying et al., 2022] as well as frameworks for more abstract glyphs [Ying et al., 2021; Jackson et al., 2018; Brehmer et al., 2021].

For an excellent review of glyphs in visualisation we refer the reader to the surveys of Borgo et al. [2013] and Fuchs et al. [2016].

2.4.3 Star glyphs

The star glyph is a well investigated glyph design that has been around for decades. Star glyphs have been used often to enable visual data comparisons [Friendly, 1991] and are still topic of active research (e.g. [Keck et al., 2017; Opach et al., 2018; Miller et al., 2019; Hu et al., 2021; Keck and Engeln, 2022]).

In a star glyph, each data point is represented as a star-shaped figure, with one ray for each variable. Up to 10 rays is common to find in the literature [Fuchs et al., 2014]. The length of each ray is proportional to the size of the variable it maps to. Star glyph are sometimes drawn with a contour outline that connects the ends of each ray. The version of the star glyph without contours is also referred to as a whisker glyph or whisker plot [Ware, 2019]. In other versions of the star glyph, the rays are omitted and only the outline is drawn. This is referred to as a 'contour plot'. These different versions of the star plot can be seen in Figure 2.9.

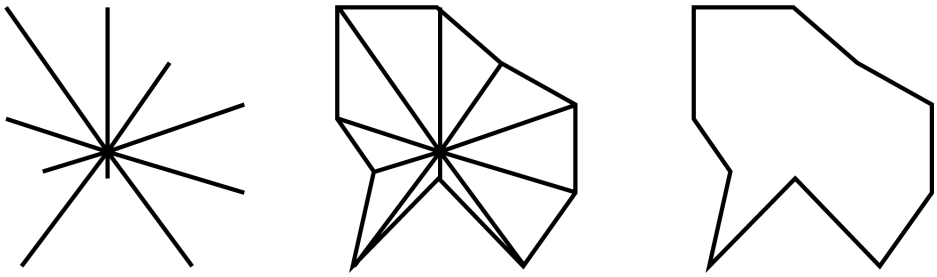


Figure 2.9: Variations on the star glyph: a whisker plot, a star glyph with outline and a contour plot respectively. Illustration based on Fuchs et al. [2014].

Task definition and measures of 'performance' vary between studies, which hinders a clear-cut comparison of the effectiveness of variations of the star glyph. For example: Fuchs et al. [2014] found whisker plots to be more effective for the comparison of datapoints, but Elder and Zucker [1993] notes that closed contour allows for faster recognition of a shape, and thus that they could be more effective in visual search. That the variance in tasks and measures for effectiveness pose obstacles for comparing visualisation approaches is a well known problem in visualisation literature [Gleicher, 2017].

2.4.4 Visual channels

When it comes to visual channels to use in glyph design, there are well researched guidelines available: many attempts have been made to order and organise visual channels by visualisation and perception researchers. Many categorisations exist, such as the seminal Bertin [1983] and its extension by MacEachren [2004]. We will, however, mainly refer to Munzner [2014], whose categorisation can be inspected in Figure 2.10. In addition, there is common consensus about pop-out effects: *colour* $<$ *size* $<$ *shape* $<$ *orientation*, in which $<$ means 'precedes' [Maguire et al., 2012; Borgo et al., 2013].

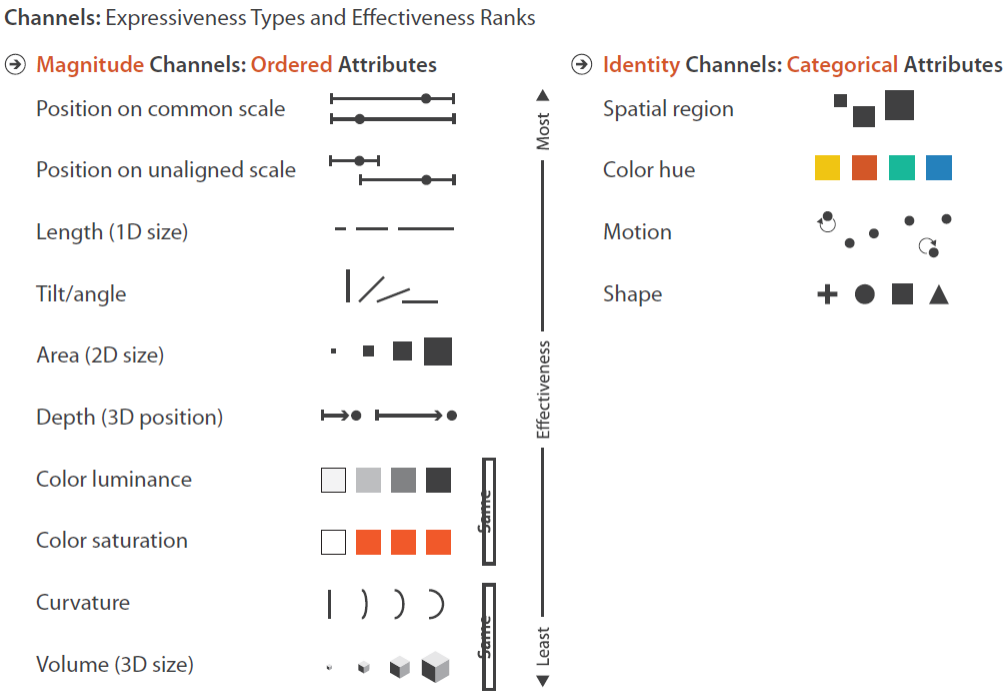


Figure 2.10: Visual channels ranked by effectiveness for ordered and categorical data, as defined by Munzner [2014].

2.4.5 Colour

The use of colour in visualisation has been researched more than any other perceptual issue in visualisation [Ware, 2019]. The general advice seems to be to be a reserved attitude towards colour: *"Do no harm"* [Tufte et al., 1990]. The literature is rife with rules on what **not to do** with colour. The important lessons on what **does** work are relatively few and described well by Munzner [2014]. We will discuss a few key aspects and concepts related to colour in the remainder of this subsection.

Colour channels

Different channels of a colour value have different interpretations: hue is categorical, whereas saturation and luminance are perceived as ordered information [Munzner, 2014]. Each of these three channels influences each others perception. Whereas hue is extremely effective to communicate categorical properties of data, the saturation and luminance as a means of conveying information on magnitude are considered much less effective than other magnitude channels, as can also be seen in Figure 2.10.

Colour spaces

Colour spaces in data visualisation are often mathematically structured colour systems [Munzner, 2014], although they can also be arbitrarily structured [Rhyne, 2017]. An extensive overview of colour spaces, their derivations and perceptual uniformity can be found in Kahu et al. [2019].

Colour maps

Colour palettes are well established in univariate and bivariate flavours [Munzner, 2014]. Trivariate palettes have been proposed too [Pham, 1990; Metternicht and Stott, 2003] and more research is being done on mapping multidimensional data to colour [Cheng et al., 2016; Ware et al., 2020]. In general however, the literature is not excited about multivariate colourmaps [Ware, 2019; Stone, 2003; Munzner, 2014].

Semantics of colour

Colour mappings can be very arbitrary, and symbolism and meaning vary through time and cultures [Morton, 1997; Ware, 2019]. On emotional interpretation the literature is inconsistent [Kress and Van Leeuwen, 2020]. For an extensive semiotic and cultural analysis of the use of colour we refer to Kress and Van Leeuwen [2020].

Colour for visual data mining

Colour has been investigated for 'visual data mining' [De Oliveira and Levkowitz, 2003] and clustering [De Runz et al., 2012; Ankerst, 2001; Ward, 2002] as well as providing insight in large datasets [Healey and Enns, 1999] and high dimensional data [Blanchard et al., 2005].

2.4.6 Curvature

In this subsection, we review how curvature has been applied in glyph design and detail the work on curvature as a visual channel.

Curvature in glyph design

Researchers have used curvature as a channel in glyph design from the 1970's on. In Chernoff faces [Chernoff, 1973], the mouth is determined by the curvature of that line. We find curvature glyphs in Weigle and Taylor [2005] and uses of curvature in flow visualisation [De Leeuw and van Wijk, 1993; Post et al., 1995]. Barr [1981] introduces the use of superquadrics in glyphs, an idea for geometric primitives of which, amongst other features, the surface curvature is parameterised. We see this design re-occurring in the literature through decades, recurring in for example: Post et al. [1995]; Shaw et al. [1998]; Cleary and Sawley [2002]; Kindlmann and Westin [2006]; Ropinski et al. [2007]; Feng et al. [2009]; Schultz and Kindlmann [2010a]; Zhong et al. [2016]; Gerrits et al. [2016], mostly in 3D glyphs, but also in 2D.

Curvature as a visual channel

Curvature is considered a pre-attentive visual stimulus [Borgo et al., 2013; Shaw et al., 1998]: a powerful way to represent data. However as a means of encoding magnitudes, it is considered ineffective [Munzner, 2014]. This is due to the compound nature of the curvature channel, which includes multiple factors such as angle, non-aligned lengths, and direction, all of which are sub-optimal channels for magnitude perception [Reynolds, 2021]. These factors magnify the difficulty of perceiving magnitude from this channel.

For categorical data, shape is considered one of the main channels [Munzner, 2014] and curvature as an attribute of shape has been applied throughout visualisation literature [Brath, 2010]. Schultz and Kindlmann [2010b] notes that a curved surface allows for the representation of three classes: convex, concave and saddles. These three classes can be translated to 2D as convex, concave and flat. Forsell et al. [2005] argues that this can be extended to five classes, as illustrated in Figure 2.11.

Figure 2.11: 5 classes that can be expressed with 2D curves according to Forsell et al. [2005].

2.4.7 Mapping data to channels

A large decision in glyph design is how features are mapped to the visualisation parameters. For example, the ideal mapping of data to star glyphs remains an open problem, for many approaches have been proposed: from algorithmic orderings to create the most symmetric and simple glyphs [Peng et al., 2004] to reinforcement learning approaches [Hu et al., 2021]. In this section, we briefly detail the different kinds of mappings and then elaborate the properties of redundant encoding, or one-to-many mapping.

Mapping approaches

According to Ward [2008], there are three different mappings:

1. **One-to-one** A one-to-one mapping assigns every data variable to a different visual channel.

2. **One-to-many** A one-to-many mapping uses redundancies by mapping one data variable to multiple glyph channels. Such a mapping can encode important variables multiple times and allow for importance-based mapping.

3. **Many-to-one** A many-to-one mapping represents multiple data variables by the same kind of visual channel, for example, the height of bars in a profile glyph.

Redundant encoding

Redundant coding, or one-to-many-mapping, is when each dimension in the data is represented by multiple of the visual channels of the glyph [Fuchs, 2015], or in other words that search can be based on any or all of the properties. Ware [2019] cites Egeth and Pachella [1969] and Eriksen and Hake [1955] to argue that, despite the fact that the amount of added value varies, there is always added value in redundant encoding. This claim is backed up by more recent studies [Nothelfer et al., 2016, 2017].

Despite Ware [2019] urging that "to make symbols in a set maximally distinctive, use redundant coding wherever possible", there has been limited research focus on designing data glyphs for one-to-many mappings [Fuchs, 2015]. And so, important questions remain open, such as how to balance dominant channels (in particular colour) with other less perceptually strong channels.

Klippel et al. [2009a] combined colour encoding with star glyphs, and observed that the use of colour increased the speed with which the users could use the glyphs. They also found that the use of colour balances out the most prominent shape features. When colour is used, redundant encoding may also serve as a fallback for colourblind users [Brunner et al., 2019; Franconeri et al., 2021].

2.4.8 Sorting Glyphs

Considering glyphs as representations of high-dimensional data, sorting glyphs is a variation of sorting high-dimensional data. Kolhoff et al. [2008] provides a sorting method based on a 1D PCA projection of the feature vectors. Although Chen et al. [2014] argues that "glyph sorting would significantly enhance the usability of glyph-based visualisation", little research has been done on this topic.

Chapter 3

Method

This chapter outlines the experimental design that we have developed for this project. We begin with a brief overview of our approach, followed by a formalisation of our methodology. We then delve into the key steps of our design process, including the feature representation selection, dimensionality reduction choices, and considerations that influenced our glyph design. Additionally, we introduce a novel user interface that builds upon our glyph design to enhance the music discovery experience. Finally, we discuss our implementation choices.

3.1 Overview

In very broad terms, we present a modified version of the system developed by Kolhoff et al. [2008]. The outline of system is depicted in Figure 3.1. While we share some similarities, such as the use of colour, we differentiate ourselves by choosing different features, requiring less user input, employing a distinct glyph design, and introducing a novel UI.

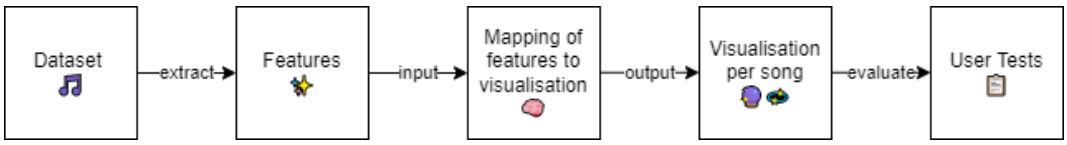


Figure 3.1: Proposed system

Building on the design of Kolhoff et al. [2008], we wanted a glyph design that would be more expressive and also planned to provide another way of mapping the parameters (Kolhoff et al. [2008] outsources that mainly to the user).

3.2 Formalisation of our design

In this section, we frame our work within a larger context of visualisation research. For this we use Munzners nested mode, as introduced in Section 2.3. We identify our work to be a problem driven work, which means that we work from the outer layer to the inside layer

in Munzners Nested Model (see Figure 2.8). In the remainder of this section, we explain our design in terms of the outer three layers of the nested model: domain, abstraction and idiom.

3.2.1 Domain

Our target users are adult users of music streaming services. They discover new music either through algorithmic recommendations, or via user-guided search and exploration. In particular, we target the users who search with an open mindset, defined by Hosey et al. [2019] as "search with no specific thing in mind, but an idea of what one would like to listen to". Search with an open mindset often starts with a 'seed song' for which similar music is sought: music with particular distinctive characteristics, as per our research question. Currently there are no visual cues for users to confirm similarity of music. Their only means of verification is to listen to each song.

3.2.2 Data Abstraction

Although we start with a raw signal, the data we finally visualise consists of derived attributes, computed from our original input signal data. The choices we make to derive the attributes are explained in Section 3.3.

Our derived attributes are tabular data, in which each row is an item and each column represents attributes. Independent attributes serve as keys, which in our case would be a song id.

Although the attributes can be interpreted categorical (with help of a linear classifier as per Spijkervet and Burgoyne [2021]), we treat them as quantitative data, normalised to a range of 0-1.

3.2.3 Task Abstraction

We identify several tasks for our visualisation, of which the main task is search. A user wants to find an unknown, or rather vaguely described, item: 'similar music', or 'music with particular distinctive characteristics'. We observe that this task overlaps with clustering of 'similar music': localising music that sounds alike is also a form of clustering these songs. Finally, we include browsing and exploring in the user's tasks.

3.2.4 Idiom

Like Kolhoff et al. [2008], we work with the idiom of the glyph. Reasons for this are the suitability of glyphs to tabular data and high dimensionality, and the fact that glyphs are very suitable for use as icons, due to their ability to convey quite some information with little pixel space available.

3.3 Latent variable representation

In this section, we will elaborate on the process of selecting the music representation or features for our experimental design. Then we describe how we explored various downstream tasks that yield latent variables that are relevant to our project. Subsequently, we explain how we selected a suitable model and describe how we validated it.

3.3.1 Downstream task

As we aim to use the representation for visualisation, we are foremost interested in the representations themselves, rather than the downstream tasks they are developed on. However, as models are first trained for a downstream task, we do have to take these into account. We distinguish the following four categories of objective downstream-tasks in music representation learning that may be of use to us:

1. **Generation** Generating music is an active area of research. Generative models typically find powerful representations as reconstructing of the data requires an effective encoding.

2. **Classification** We hypothesise classification tasks are useful to us as they often rely on representations of other high-level concepts. In addition, they are well suited for clustering and a very active area of research.

3. **Recommendation** We hypothesise that our task of retrieval/identification is related to recommendation and that representations for recommendation capture characteristics that may benefit our task.

4. **Multi-task** As with generative models, the aim here is to obtain a representation that generalises to many tasks and captures semantically important factors [Kim et al., 2020].

3.3.2 Model selection

Based on the categories defined above, we collected methods for content-based MIR representation learning that may be suitable for our project. For this, we looked at the state of the art research and reviewed the latest ISMIR papers. Given the difficulty of training (both in obtaining data and resources required), we restricted our exploration to papers that provide full code and trained weights.

We had an in-depth look at 23 papers, of which 10 provided pre-trained models. We then looked at their respective downstream tasks and performance to order the models on suitability. Although we intended to include a model for the downstream task of recommendation in the list, we did not find one that fit our purposes: the recommendation-based representation learning models we considered were either developed in a multi-modal fashion [Martín, 2017; Chen et al., 2021] or do not provide code or weights [Saravanou et al., 2021; Van den Oord et al., 2013]. The resulting list can be found in Table B.1 (Appendix B).

The model on top of the list was Spijkervet and Burgoyne [2021]: Contrastive Learning of Musical Representations (CLMR). The CLMR model is an adaptation of the very effective SimCLR model [Chen et al., 2020], which was developed for contrastive learning of visual representations. Contrastive learning an unsupervised representation learning technique for representation learning in which the objective is to maintain similarities and dissimilarities between datapoints in the representation space. We considered CLMR a very suitable model as it not only recently achieved state of the art in representation learning for MIR, it is also well documented and lightweight to run. The network has learned a representation of 512 dimensions over an input sample of about 2.6 seconds (59049 samples at samplerate 22050). The model was trained for the downstream task of classification on the MagnaTagATune dataset [Law et al., 2009]. We use the model and weights as provided by the authors. Representations over longer segments (we use up to 30 second samples) are averaged.

3.3.3 Verification of representation

In this section, we verify the effectiveness of the CLMR representations by examining feature clusters to genre classification on multiple datasets and comparing feature embeddings to the Spotify features. Our results demonstrate the model's strong performance across various datasets and tasks, leading us to select it for use in our experimental design.

Comparing feature clusters with genre labels for 3 different datasets

First, we compared how the derived features behave for the same task of genre classification on different datasets. The datasets we used for this are MagnaTagATune, GTZAN [Tzanetakis and Cook, 2002] and a small custom dataset that we constructed for this test. All three of these datasets provided genre labels as ground truth. We extracted the features for samples of each of the datasets, embedded them in 2D space with the UMAP algorithm and colour-coded the datapoints according to their genre labels. We find that clusters in the UMAP embedding tend to correspond with clusters of genre labelling, which we interpret as a clear indication that the feature space captures - in an abstract way - features that correspond to meaningful high-level concepts. We note our findings on the different datasets here and refer to Figure B.1 (Appendix B) for plots of the 2D embeddings.

- **MagnaTagATune** We extracted features for the MagnaTagATune test split as defined by Spijkervet and Burgoyne [2021], to ensure this is indeed unseen data for the model. We observe noticeable overlap between genres that also have similarities. For example new age and ambient are both pretty slow electronic music with a lot of synthesizers and few vocals. Rock and metal also overlap, both genres that are guitar and drum heavy. See also Figure B.1a,

- **GTZAN** The GTZAN dataset provides 100 samples for 10 genres each. The clustering was less well defined in 2D but there is a well defined distribution visible in the 3D UMAP. GTZAN has noted flaws [Sturm, 2013] such as repetitions, mislabel-

ings, and distortions that may explain why it was a bit harder to find clusters in the embedding. See also Figure B.1b.

- **Custom dataset** A small custom dataset was constructed from several albums that are considered iconic for several genres, as listed in Table B.2 (Appendix B). Remarkably, the small dataset of only 11 albums demonstrated noticeable clustering, as can be seen in Figure B.1c.

Comparing feature clusters with Spotify features

In addition to their performance on genre classification, we wanted to see how the representations perform in contrast to the Spotify features (see Section 2.1.4). For this purpose, we constructed an additional dataset of 10.000 datapoints. To construct this dataset, 10.000 feature entries were randomly selected from the 1.2M+ Spotify Dataset by Figueroa [2020] and 30 second mp3 samples for each corresponding song were scraped via the Spotify API.

We extracted features for all 10.000 samples with the CLMR model, embedded them in 2D the UMAP algorithm and colour coded the datapoints according to the value of a Spotify feature for that sample. We found that the representation clearly picked up relations that correlate with the Spotify features such as danceability, energy, valence, loudness, acousticness and instrumentalness. This can also be seen in Figure B.2 (Appendix B).

3.4 Dimensionality reduction

In this section, we explain our decisions regarding dimensionality reduction. We provide a comparative analysis of several methods and describe our selection process for the most appropriate method.

The CLMR model provides a representation of 512 floating point values. In literature, we observe consistently that a lower number of dimensions used in star glyphs, the more effective they are for a variety of tasks, in terms of accuracy and time required [Fuchs et al., 2014; Dy et al., 2021; Hou et al., 2022]. As we wanted to both preserve the richness of the embedding we settled on a 8 dimensions, which also works well with the axis arrangement we chose (see Section 3.5.4).

We were particularly interested in a dimensionality reduction method that preserves clusters. Therefor, we evaluated how 5 algorithms preserved the similarity between data points: PCA [Pearson, 1901], t-SNE [Van der Maaten and Hinton, 2008] and UMAP [McInnes et al., 2018] and two algorithms that propose improvements upon the latest achievements of t-SNE and UMAP: TriMap [Amid and Warmuth, 2019] and PaCMAP [Wang et al., 2021]. We used default settings for all algorithms in this first search.

To compare how the similarity between vectors in the 512 dimensional space were preserved in 8d space, we calculated the cosine similarity matrix for both the 512 dimensional representations, and the 8d embeddings. We interpret the rows in both matrices as 'similarity vectors' for each datapoint, expressing similarity with all other vectors. To investigate this, we calculated the pairwise cosine similarities between the row entries in the similarity matrices. The resulting statistics can be found in Table C.1 (Appendix C).

We found that UMAP is, statistically, competitive with PCA. As PCA is a linear transformation and the other algorithms do not preserve linearity, it is not surprising that it performs well. However, the clustering effects of PCA dimensionality reduction are limited. As both TriMap and PaCMAP performed rather poor (despite their boasting about the little parameter tuning required), we decided not to investigate them further. However, we found ourselves surprised by the bad performance of t-SNE as well as curious for the effects of hyperparameters with UMAP. Therefor we conducted a hyperparameter search for both t-SNE and UMAP.

For t-SNE, we explored the settings of the 'perplexity' hyperparameter for values between 5 and 50, as recommended by Van der Maaten and Hinton [2008], and also included larger as the documentation of open t-SNE recommended higher values for larger (10.000 datapoints) datasets [Poličar, 2020]. All calculations were done on a 50D PCA projection of the original data. The statistical results can be found in Table C.2 (Appendix C). We observe consistent poor relations between the similarity vectors as constructed by the t-SNE projections. The explanation is most likely to be found in the implementation and the fact that t-SNE is by no means optimised for 8d projections: t-SNE is usually optimised for 2 or 3 dimensions [van de Ruit et al., 2021]. To achieve better results, the algorithm would likely have to run for days.

For UMAP, we explored the settings of the min_dist and the nearest neighbours parameter settings. The results can be seen in Figure C.3 (Appendix C). We observe that the algorithm seems remarkably robust to hyperparameter settings. The UMAP algorithm also does not need the kind of optimisation t-SNE requires at all, which makes it more suitable for this kind of dimensionality reduction. So due to the high preservation of similarity between datapoints, its relative fast calculation times and clustering capabilities, we decide to keep using the UMAP embedding technique. For the applications in this work, we used a nearest neighbours setting of 15 and minimum distance of 0.2. The embeddings obtained after the UMAP transformation are normalised to a 0-1 range.

3.5 Glyph design

In this section, we present our novel glyph design which aims to enhance the expressiveness of music representation and provide an alternative means of mapping parameters compared to the approach adopted by Kolhoff et al. [2008]. Our design incorporates several features, including the use of colour, redundant encoding, dimension ordering, and curvature, all of which we will detail in this section.

3.5.1 Star Glyph

For our glyph design, we first consulted the literature (see also Section 2.4). Few guidelines are available on what glyph design works best for visual search [Fuchs, 2015]. Eventually, our eyes landed on the star glyph: the star glyph has proven a popular method to enable visual data comparisons, it is simple yet versatile and capable of expressing a wide range of shapes.

After much deliberation, we decided to work with the contour plot variant of the star glyph for its superior expressiveness, as opposed to the less expressive whisker plot. We also recognised the significance of shape as a feature, as demonstrated by Palmer [1999]. The inclusion of a contour improved the expressiveness of the icon, while we chose to omit the rays to achieve a simpler design.

We further adjusted our contour plot to use dual colouring by adding an 'inner' and an 'outer' colour to an 'inner' and an 'outer' shape, as explained in Section 3.5.2. We also added curvature to the line segments, as explained in Section 3.5.5.

3.5.2 Dual colour

In visual search, we want to take advantage of the strong visual channel that is colour. After all, Ware [2019] urges to "use strong pre-attentive cues before weak ones where ease of search is critical". Having dutifully observed that the use of colour is usually reserved for categorical data [Munzner, 2014], we choose to follow Kolhoff et al. [2008] in their automatic mapping of colours: they mapped 6 of their features to 2 RGB colours: an inner and an outer colour. As the shape of their icon was not very expressive (always circular and symmetric), we suspect that the clustering properties they found for their icon depend largely on the colouring approach they used.

After ordering the dimensions according to their variance, we map the first six of our channels to RGB colour space. The two resulting RGB colours are used as an inner and an outer colour in our icon. The gradual changes that are allowed for by the dual colour, allow in our opinion for a suitable overlap between music categories; after all, these rarely have strict boundaries. To illustrate this, we have plotted the top 10 predicted labels of the dataset against the a circular display of our colours in Figure D.1 (Appendix D). We observe that similar clusters emerge.

To be able to use the 'inner' and 'outer' colour, we work with an 'inner' and 'outer' shape, which ideally should be balanced in area. As we want the glyph to be clear, both shapes should be pronounced. We considered several methods to construct the outer shape based on the inner shape, amongst which distance fields, but found those to distort the shape too much. Eventually we settled on a fairly straightforward method: we first draw the 'inner' shape based on the feature vector, then extend 0.5 along each axis and re-scale all lengths by 0.8 to slightly reduce their range (and so to balance the area of the outer shape). These values are used to draw the outer shape. The scaling values were mostly determined empirically and could possibly benefit from a more algorithmic approach.

3.5.3 Redundant encoding

In the literature review on glyph design (Section 2.4.7), we have outlined the potential advantages of redundant encoding. These benefits include increased expressiveness, improved distinctiveness, enhanced visual search capabilities, and greater resilience to colour blindness. To this end, we have proposed a variable mapping scheme that encodes each variable twice: not only as a parameter of shape in the contour plot, but also influencing colour

or curvature of the shape. To which degree the expectation of a colour blind-proof design turned out correct is evaluated in Section 5.4.

Our proposed parameter mapping can be seen in Figure 3.2. To demonstrate the influence of each parameter, we refer to Figure D.2 (Appendix D), where we set all parameter values to 0.5, and vary one parameter at the time.

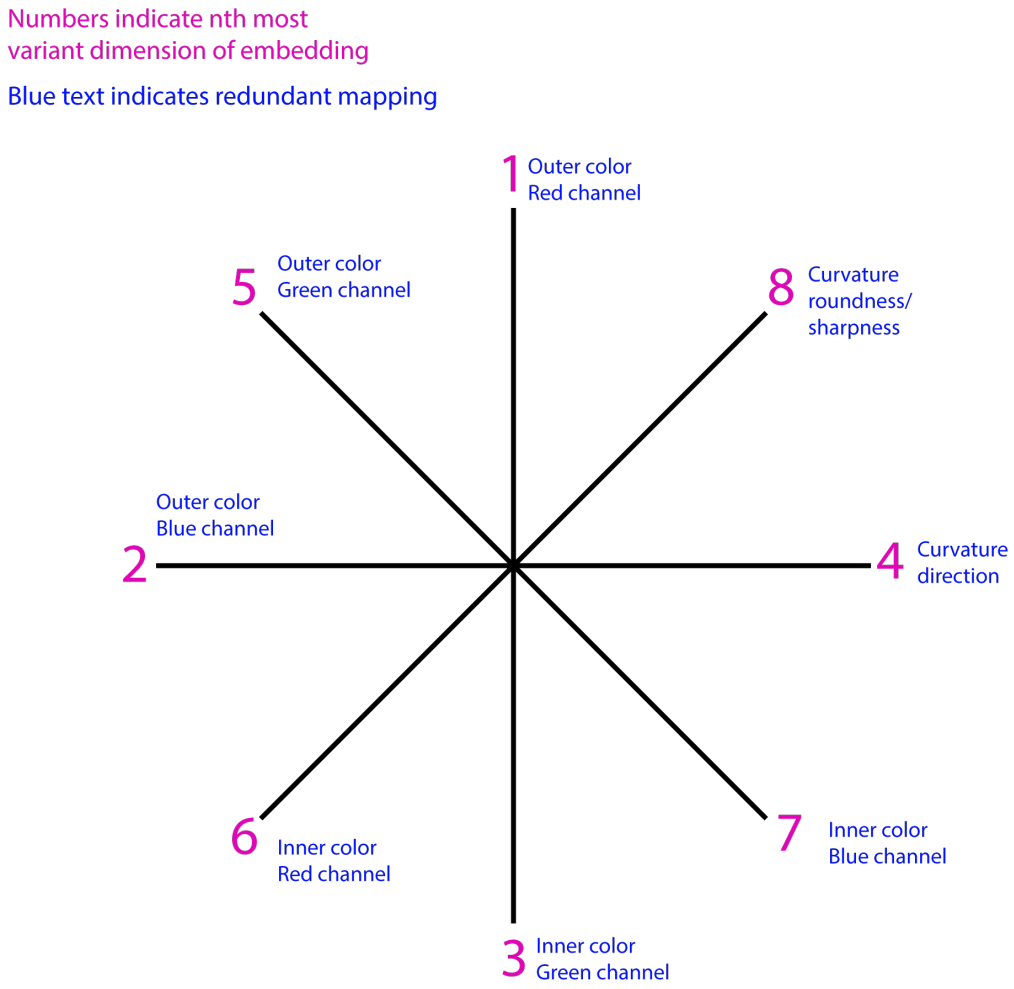


Figure 3.2: Proposed variable mapping along the axis of the star glyph.

3.5.4 Dimension ordering

The order in which variables are mapped to the axes of the star glyph have a big influence on the resulting shapes. The same data-point, when mapped in different orders, can be displayed as very different shapes, as is illustrated in Figure 3.3. The ideal order is considered an unsolved problem, and may depend on the task at hand, but we take an interest

in the work of Klippel et al. [2009b], which investigated the ordering of star glyphs and the mapping of their most varying dimensions. They found that the most 'salient' shapes were faster to detect. They thus recommend to arrange the rays such that the main variation occurs along the main axes, in the case of eight variables.

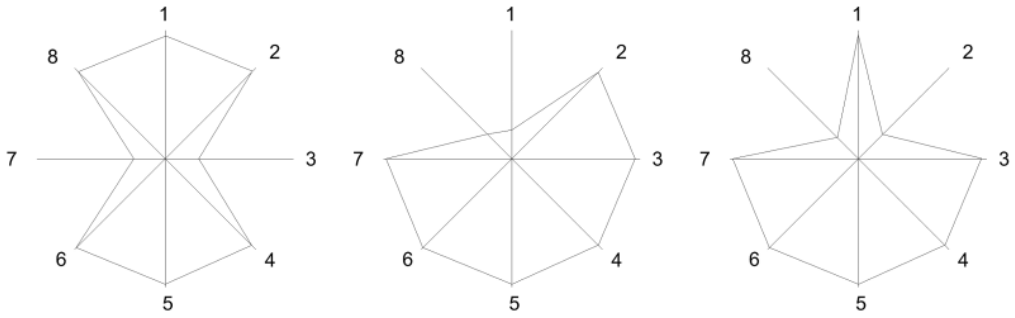
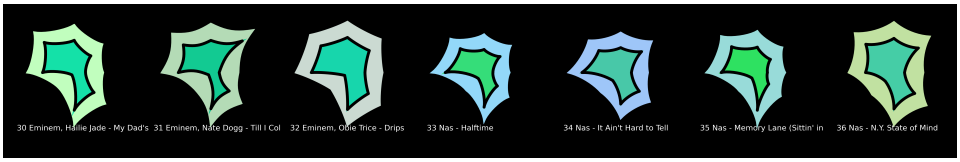


Figure 3.3: The same data mapped to a star glyph in different orders. Image from Klippel et al. [2009b].

When trying this out, the results were more than interesting: it seems that per genre, more variance is visible in both shape and colour, especially within genres. See Figures 3.4 and 3.5 for a comparison of the ordering of the dimensions. We can observe how local contrasts are much more clearly distinguished. In the sorted version, more variance in shape and colour can be seen, especially within genres.

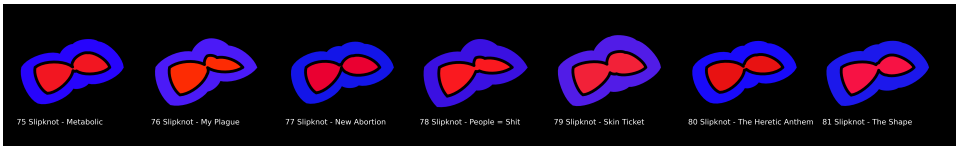
In Figure 3.4a, we see the icons for seven rap songs without sorting the axis. They all have a similar shape and the same inner colour. In Figure 3.4b, after sorting the axis as proposed by Klippel et al. [2009b], we observe much larger variance in both shape and colour. In Figure 3.5a, we see the icons for seven heavy metal songs without sorting the axis. It is very hard to detect differences between them. In Figure 3.5b, after sorting the axis, we *can* detect small differences between the icons.



(a) Subset of icons from test set, based on 8 dimensions, unsorted

(b) Subset of icons from test set, based on 8 dimensions sorted on variance and aligned to x-y axis

Figure 3.4: Effect of sorting the axes of a our glyph on variance



(a) Subset of icons from test set, based on 8 dimensions, unsorted

(b) Subset of icons from test set, based on 8 dimensions, sorted on variance and aligned to x-y axis

Figure 3.5: Effect of sorting the axes of our glyph on variance.

3.5.5 Curvature

We consider the powerful pre-attentive properties of curvature to be a compelling reason to work with it. We expect it to broaden the range of possible shapes and therefor to increase the expressiveness of the icon. Klippel et al. [2009b] has observed however that 'salient shape characteristics' increase the classification speed, but also warns that salient shapes *can* introduce a perceptual similarity so strong that it overrides assumed similarities in data. We speculate that curvature of the outlines of the shape of our glyph may provide balance for the strong channel of colour.

Using the first six channels for colour, we mapped the 7th and 8th dimension to the curvature channel in the following way: The 7th dimension, the axis with the 4th largest variance, is set to determine the direction and strength of the curvature. The 8th dimension, the axis with the least variance, is set to determine the distance of the control points from the endpoints of the line segments. An illustration of the influence of these two parameters on a curve can be seen in Figure D.3 (Appendix D).

The clear drawback of this method is that the eighth dimension is not visible when the seventh is (close to) zero and in such cases, its overall impact may not be very clear. As the eight dimension is the dimension with the least variance, and it is still encoded in the overall shape, we consider this an acceptable choice. An illustration of this singularity can be seen in Figure D.4 (Appendix D), where the effect of the curvature settings can be viewed on an 8D star shape.

We restricted the range for the direction and strength of the curvature to [-0.3, 0.3] and the distance of the control points between [0.2, 0.6]. We chose not to make the angular curve type too edgy, or any resulting shape may be mistaken to consist of more line segments than it actually does. To prevent unwieldy intersecting curves, we apply an intersection detection method. When two segments intersect, we iteratively loosen the strength of the curvature until they no longer intersect.

3.6 UI

In this section, we introduce a new search interface based on our method, a method to enhance contrast between icons and a sorting mechanism that can make the use of our icons in a playlist more effective.

The UI components have been implemented in a web app and can be tried at `musicons.io`, which we strongly recommend.

3.6.1 Search-by-icon

We propose a novel search method, in which the user can adjust the icon to the kind of music they are looking for. This is much like a reverse search proposal and is conceptually related to Knees and Andersen [2016], who proposed audio search by drawing the mental images of sound.

The user is presented with 8 sliders: one for each parameter of the icon. The icon is real-time updated when dragging the sliders, giving the user opportunity to explore the possibilities of the icon and what music it represents.

The search is implemented by finding the icons that have largest cosine similarity. A comparison of the vector based on the slider values with all datapoints in a dataset of 10.000 songs is done each time a user is done dragging a slider. The resulting top 10 songs are displayed to the user.

We have also tried doing search based on the position of the datapoints in a 1D UMAP embedding of the 8D space. However, the available JavaScript implementation of UMAP is resource intensive and the random seed for the projection of new datapoints cannot be fixed. This means that even though the larger 1D embedding can be pre-calculated and the results of this approach do yield a high average cosine similarity, the same target embedding will yield different results each time, which vary much too widely and make for a confusing and unsatisfactory experience as it is impossible to retrieve earlier results. Therefor, we have decided to stick to cosine similarity, as that yields consistent results for the same icon, makes perceptually more sense and can be calculated very fast too.

3.6.2 Enhancing contrast

The embedding used was generated over 10.000 samples, across many genres. This makes differences between genres in the colouring larger, but differences in between (sub-)classes smaller. To maximise local differences, we implemented a means of re-scaling the parameters of a set of icons (for example a playlist). For this we used the min-max scaling approach, and re-normalises the features of the selected set to the 0-1 range. In addition, we provide the user with the possibility to linearly interpolate between the embeddings and their re-normalised versions. This allows for a gradually increasing contrast.

We expect this to be useful when looking at playlists with music that is very similar. The results of this implementation are discussed in Section 4.6.

3.6.3 Sorting

Two sorting methods, 1D and 2D, are provided by Kolhoff et al. [2008], both based on a PCA of the icon parameters. We tried several approaches on the dataset, and found that UMAP embeddings once again outperformed PCA, as is illustrated in Figure D.5 (Appendix D). Moreover, on initial examination, UMAP embeddings on icon parameters seemed to work better than on the 512 dimensional space, which is in line with Kolhoff et al. [2008]'s approach, but using more advanced methods than PCA. We also implemented UMAP sorting in the user interface, and it worked best when applied to the 8D embeddings of the playlist, rather than using the entire 1D embedding of the dataset and selecting the closest ones. We believe this sorting approach can make the use of our icon in a playlist setting more effective.

3.7 Implementation

The development of the icon design was first done in python for its suitability to working with high dimensional data, dimensionality reduction and the possibilities it offers for rapid prototyping. In our python implementation, we used the following libraries:

- Sklearn for PCA, t-SNE and k-means clustering
- UMAP-learn for UMAP (random seed: 1989)
- CLMR codebase by Spijkervet and Burgoyne [2021]
- Matplotlib for plotting the glyphs, with own bezier curve intersection detection

We calculated the UMAP embedding based on the features we extracted for the 10.000 song Spotify dataset (the dataset we described in Section 3.3.3). The resulting dimensionality reduction model, which we used to transform all the features into 8D space. Any additional songs that were included later for user testing purposes were also transformed using the UMAP model that had been fit to this dataset.

Throughout Chapter 4, Chapter 5, and our online demo, we consistently employed the same UMAP model fitted to the representations extracted from the 10.000 song Spotify dataset. Due to the stochastic nature of the UMAP algorithm, we acknowledge that to some extent these resulting icons are arbitrarily assigned: for any other initialisation (a different random seed), we would have obtained different results as the orientation could have ended up totally different. However, we maintain that analysing such results would, despite different appearance, still yield similar clusters and properties.

After finalising the icon design, we rewrote the rendering in JavaScript so the icon could be easily integrated in an interactive UI. We wrote the UI in a web app to facilitate remote user testing. The web app loaded the 8D feature vectors that were calculated in python previously and stored in JSON files.

Once the icon design was finalized, we proceeded to rewrite the rendering process in JavaScript, making it easier to integrate the icon into an interactive user interface (UI). To facilitate remote user testing, we developed the UI as a web application. This web app was

designed to load the pre-calculated 8D feature vectors, which had been previously computed in Python and stored as JSON files

For the web app, we used the following libraries:

- React for state management and flow
- Canvas Web API for drawing
- PaperJS for bezier curve intersection detection
- UMAPJS for 1D sorting
- Bjorn Lu's 'Colorblind' for the colour blindness simulation

Our web implementation allows for real-time search in a database of 10.000 songs. We have taken a brute-force approach in the search-by-icon method and not tested for an upper bound on the possible number of songs. We suspect, that the number of songs can be much higher, especially with optimised implementation.

Chapter 4

Visualisation results

In this chapter, we will highlight some of our observations on the results of our method: First we have a look at the results of our icon design, then go over its clustering effects, followed by the possibilities of outlier icons. We then have a look at our novel Search-by-icon UI, playlist sorting and finally: increased contrast.

These results can also be seen in the online web application that was made for this project, where you can listen to the matching audio too. We strongly encourage you to inspect the results there for yourself: `musicons.io`. In Chapter 5, we evaluate our design with a user study.

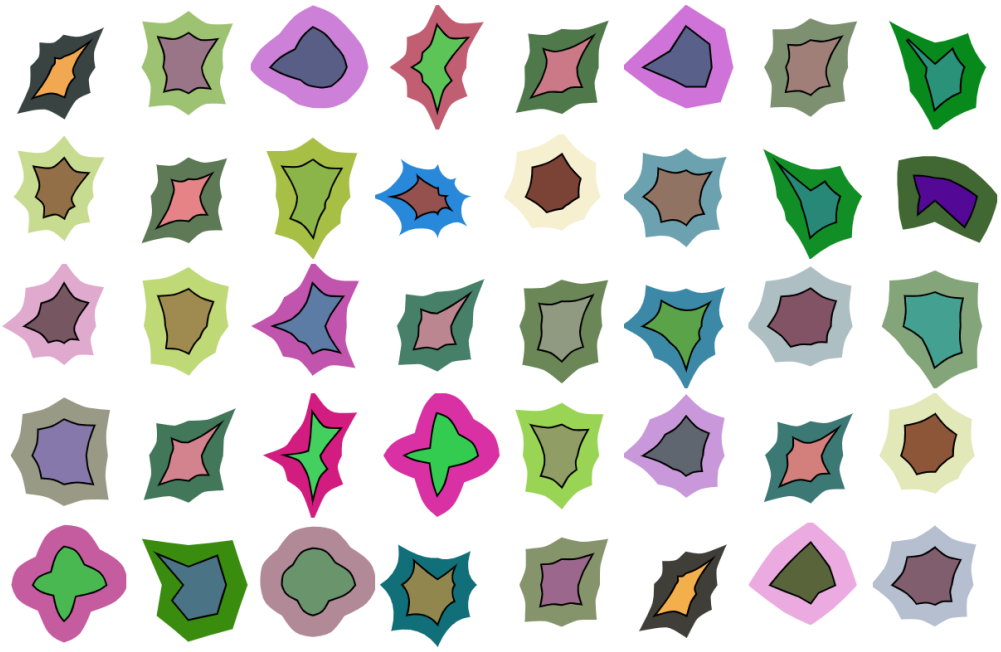


Figure 4.1: 40 randomly selected icons from the results

4.1 Icon

For a first impression, we randomly selected 40 icons from the results in Figure 4.1. We observe that the icon is, as intended, very expressive and covers a broad range of shapes and colouring. We observe that the colours are rarely very saturated. This may be a result of the features being re-scaled to the 0-1 range, which might makes the number of points that have very high values for colour channels a small minority.

The icons in Figure 4.1 are set against a white background, but as the icons were developed to use in a dark Spotify-like environment, the screenshots in the remainder of this chapter will show them against a dark background.

4.2 Clustering effects

What we were hoping for, and had suspected based on the clustering in 2D embeddings, namely similar icons for similar music, seems to happen at a first glance. In Figures 4.2, we show some screenshots from playlists that were comprised of multiple genres. When inspecting the playlists, we found some very neatly clustered icons, as seen in the screenshots.

(a) EDM (b) Gospel (c) Indie rock

Figure 4.2: Icons for songs of different genres seem to cluster well.

We observe that some clusters seem more 'finegrained' than others. Whereas the songs in Figure 4.2 all very much resemble the songs of the same genre, we observe something

else in classical music: In Figure 4.3, we see icons generated for a playlist of calm classical music. We observe, in the colouring, strong distinctions between both shape and colour of songs that contain piano or violin instruments. Violin music seems associated with green icons, with one prominent spike on the diagonal toward the top-left. Piano music, on the other hand, is characterised by blue icons that are more pointy. It seems that the features are rather sensitive to the timbre difference between the two. Most interesting is the last song (number 12), which contains both violin and piano partitions, and it is also visually in between the 'piano' and 'violin' icons.

Figure 4.3: Icons for 'calm' classical music with violins and piano partitions.

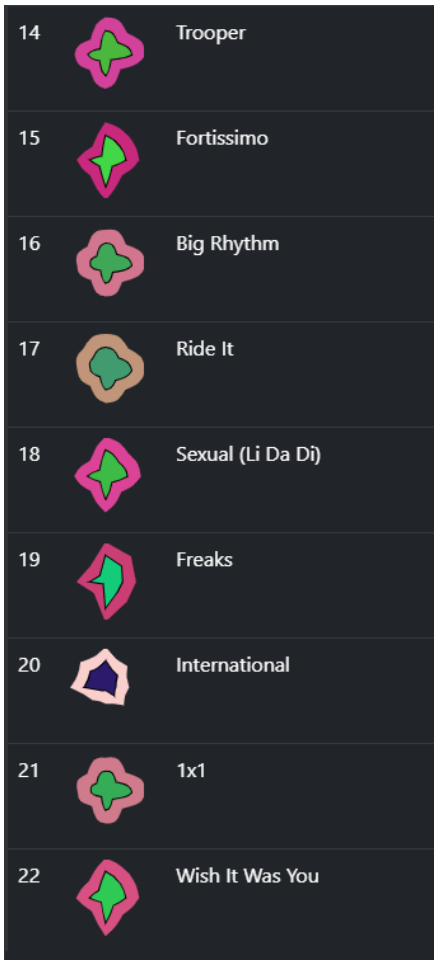


Figure 4.4: Outlier icon between EDM songs (song 20).

4.3 Outliers

We also find apparent outliers. Sometimes this is because the sample is cut from an unfortunate part of a song, or because the song itself not what we had expected it to be. However,

sometimes we find that a song sounds rather similar to other but still deviates visually. For example, in Fig 5.16: while most song in this list of Electronic Dance Music (EDM) songs are characterised by a pinkish outer colour and a green inner colour, song 20 looks rather different. With its dark purple inner colour it seems more reminiscent of one of the gospel songs in 4.2. If we were to classify the song however, we would assign it to EDM but we do note that song 20 **does** sound different from the songs with the bright pink/green icons that surround it, as the base of the other songs is much much heavier than the bass of song 20. In addition, song 20 also has more 'classical' instruments integrated such as guitar and even a little bit of bagpipes. It may very well be that the model is responding to these different timbres.

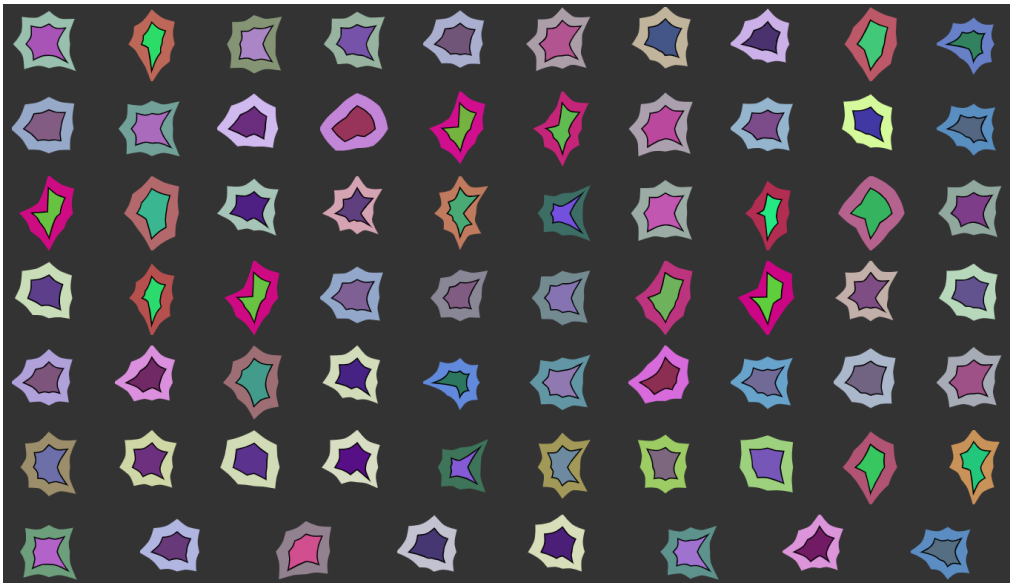
4.4 Search-by-icon

The UI for Search-by-icon can be seen in Figure 4.5. The icon parameters are indicated with a dotted line and the icon is updated real-time when the parameters are changed with the sliders. While the users drags the slider, the most similar songs are immediately updated. We find that the icon does not have to be a perfect imitation to retrieve music that seems rather agreeable with the icon.

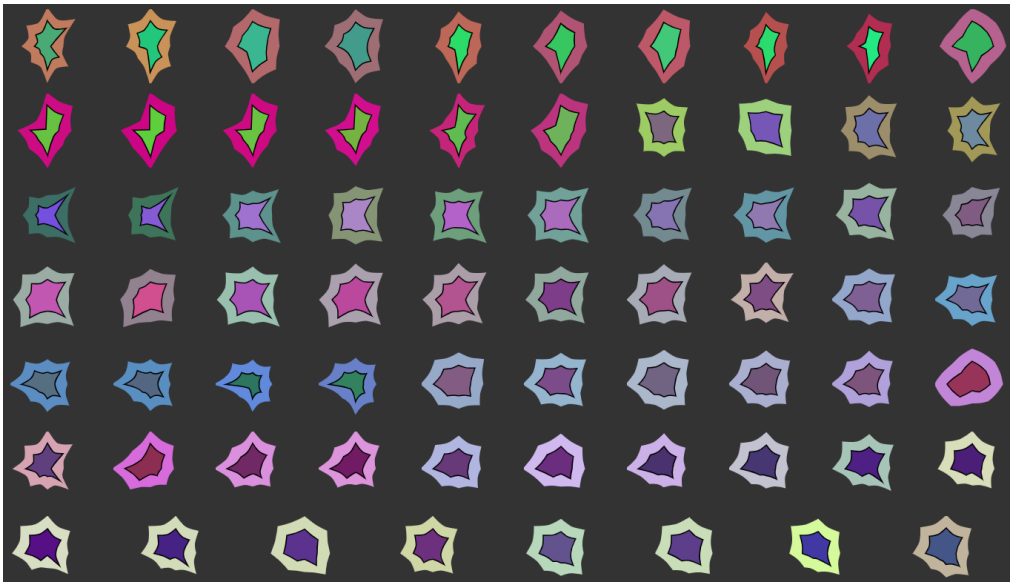
Figure 4.5: UI for Search-by-icon.

4.5 Playlist sorting

In Figure 4.6 an example of the effect of playlist sorting can be seen on a rather diverse playlist. We observe that each sorting will yield a different order and that it may not be perfect, but perceptually close items seem to be positioned withing close proximity of each other.



(a) Before sorting

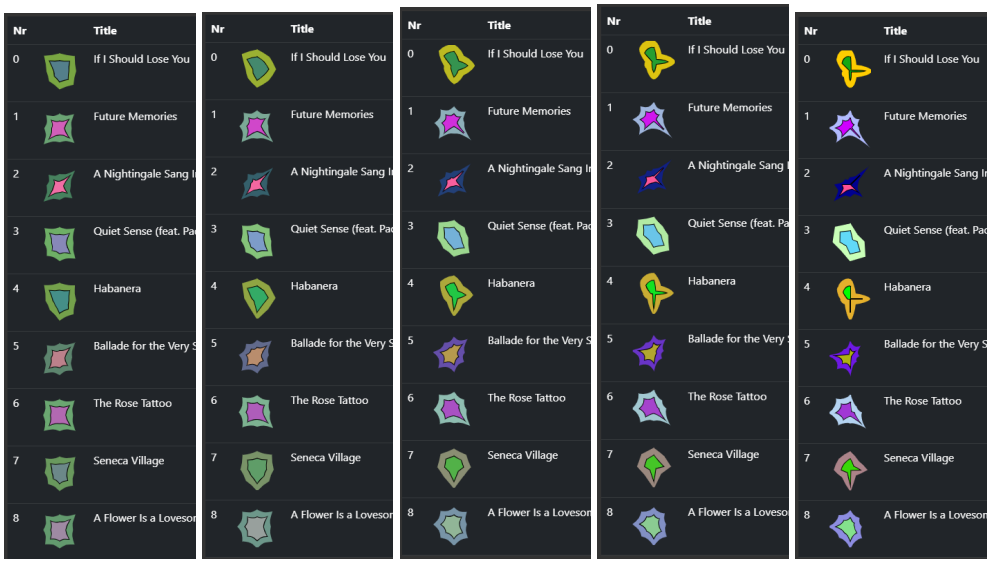


(b) After sorting

Figure 4.6: Effect of sorting on playlist, order: left to right, top to bottom.

4.6 Playlist contrast

The result of increasing contrast by min-max scaling the features of a subset of the icons can be seen in Figure 4.7. In this Figure, we observe find a small sets of datapoints that are rather similar: all jazz songs with a sort of green-ish icon. By increasing the contrast we can more easily find which icons are most similar to each other, such as song 0 and 4, or song 1 and 6.



(a) Contrast: 0% (b) Contrast: 25% (c) Contrast: 50% (d) Contrast: 75% (e) Contrast: 100%

Figure 4.7: Rendering icons for a small playlist with jazz music with increasing local contrast.

Chapter 5

Evaluation

Due to the subjective nature of visualisation and perception of music, we evaluate our hypothesis and the consequential experimental design with user tests. To verify to what degree our solution is an answer to the main research question, we have performed five user tests. This chapter explains the overall setup of the user study and then details per test its objectives, the selection and presentation of stimuli, an analysis and an interpretation of their results. For a detailed and in-depth discussion of our findings, please refer to Chapter 6.

5.1 Study setup

In this section we outline the general setup of the user study: its outline, setting, instruction and panel composition.

For a full evaluation, we designed five tests that allow for the evaluation for different parts: from the effectiveness of the icon to a larger system-evaluation. Although we have addressed all three sub-questions in this thesis project, the first two have been answered largely with literature reviews and experimental validation of existing models. The novelty in our work lies in designing a new icon and mapping it to latent variables, rather than a new feature extraction method for music. Therefore, the emphasis of this evaluation will be on the icon design and its utility in a system, rather than its features' mapping to music.

We specify the design of our user study according to Munzners nested model, in particular addressing the 'idiom' and 'abstraction', as illustrated in Figure 5.1. The first three tests are concerned with idiom, the last two with abstraction.

5.1.1 Outline

The user study consists of five tests, each of them based on tasks as described by Cunningham and Wallraven [2011]:

1. Test 1: Clustering (free-grouping task)

2. Test 2: Outlier Detection (5-alternative forced-choice task)

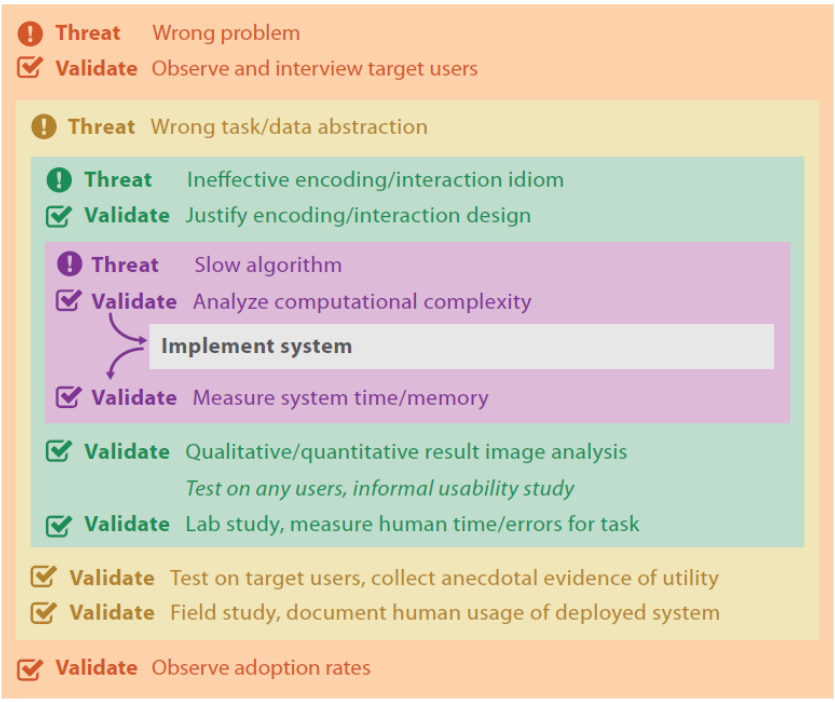


Figure 5.1: Munzners Nested Model, listing validation methods per layer. Image from Munzner [2014]

3. Test 3: Generalisation, Contrast and Colourblind Robustness (matching-to-sample task)

4. Test 4: Search-by-icon (real-world task)

5. Test 5: Search-in-playlist (real-world task)

Note that the first three tests are testing the properties of the icon, our chosen idiom. These tests resemble tests that might be performed in a lab study. Both test 4 and 5 are concerned with tasks that a participant might encounter outside of a lab study: using our novel search method and the application of the icon and its sorting properties in a playlist-setting.

Test 1 and 2 are, together, a replication of one of the experiments in Kolhoff et al. [2008]. We consider these useful tests in themselves, and also think the possibility to compare our project to its predecessor has merit.

5.1.2 Stimulus selection

To minimise bias in the design of the tests, we selected the stimuli used in the test algorithmically from the 10.000 Spotify song dataset. We partitioned the dataset into clusters by applying the k-means clustering algorithm on the 8D feature vectors of the songs, using 10 and 40 as values for k, respectively.

We found that for k = 10, the clusters are clearly partitioning the icons into groups but that there is still a large variety within these groups. For k = 40 we find that the groups are, as expected, much more fine-grained and sometimes even surprisingly homogeneous. Some of the resulting clusters for both values of k can be seen in Figures E.1 and E.2 (Appendix E).

Per task, we then randomly sampled from the clusters. This enabled us to ensure diversity in the selection, or similarity, when necessary. For tests where a diverse selection was important, for example to set up a diverse playlist, we would sample from each the k = 10 clusters. That would yield us confidence that we had sampled from large parts of the latent space. For tests where similar datapoints were required, for example in a matching-to-sample task test, we sampled from the clusters we found with k = 40.

5.1.3 Order

To mitigate the influence of order effects in this study, we have implemented randomisation where possible. By doing so, we aim to exclude the possibility that the participants' responses we observe are affected by the order of conditions to which they were exposed. We have randomised the presentation order of stimuli, rendering modes, the selection of targets and the sequence of tests where possible.

5.1.4 Participants

We define the population as "non-experts, but people who are in general computer-literate" adults [Cunningham and Wallraven, 2011]. In selection, we aimed to be diverse in terms of gender and age. We aimed for a roughly equal amount of men and women and an equal distribution of ages in 20-29, 30-39, 40-49, 50-59, 60-69 and ¿70. Participation was fully anonymous to minimise response bias.

The size of the panel was calculated with an a-priori sample size calculator [Dhand and Singh]. Anticipating a medium effect (d = 0.5), we find that we require least 27 participants to maintain a power level of 0.8 and probability level 0.05, on the premise that we validate our significance with a paired samples t-test (as is the case for Task 3 and Task 4B).

Aiming for at least 27, we collected 38 responses. The distributions for their respective age, gender and experience with the Spotify streaming service can be seen in Figure E.3.

5.1.5 Instructions

As per the recommendation of Cunningham and Wallraven [2011], participants were provided written instructions to the participants for each task, which they could retrieve from the menu bar at any time.

5.1.6 Setting

To perform the study, we developed a web app that presents the user with the instructions, facilitates the flow and timing, randomises orders and collects the data. The app was ac-

cessed remote: we were not to present when participants did the user study, to minimise response bias as per the recommendation of [Cunningham and Wallraven, 2011].

The obvious drawback of the remote setup is the lack of understanding of what happened during a test, for example if a participant had been disturbed or taken a break. Therefor the web app logged the actions of users, such as clicks and other interactions with the elements of the web app. This helped us explain large aberrations in results. For example we could explain very long in time-on-task, when it turned out a participant had taken an hour long break. Or in another setting, we could check if participants had understood the task by verifying if a participant had even listened to all a song before labelling it as 'most dissimilar'.

In the remainder of this chapter, we will go over each of the tests in detail, explain their goal, stimuli selection and presentation and analyse and interpret their results.

5.2 Test 1: Clustering

This is the first test that is performed the study in Kolhoff et al. [2008], we discuss the second test in subsection 5.3. It is a classic 'free-grouping' or 'free-sorting' task, used a lot in the discipline of psychology [Blanchard and Banerji, 2016].

5.2.1 Goal

The goal of this test was to evaluate how well the icons capture the 'similarity' of their features and how much users agree on this.

5.2.2 Description

Participants were asked to visually form clusters from a set of 60 icons, without knowing song titles or other information. Participants could use any number of clusters and were allowed to leave a set of spare icons that did not fit to anything else.

5.2.3 Stimulus selection and presentation

For the stimuli, we sampled randomly from the clusters mentioned in Section 5.1.2. To ensure that there is a diversity in the selection yet still the possibility to make clusters, we sampled 10 datapoints from 6 of the clusters that were found for k = 10. Participants were thus presented with the icons for 60, which is close to the number Kolhoff et al. [2008] used.

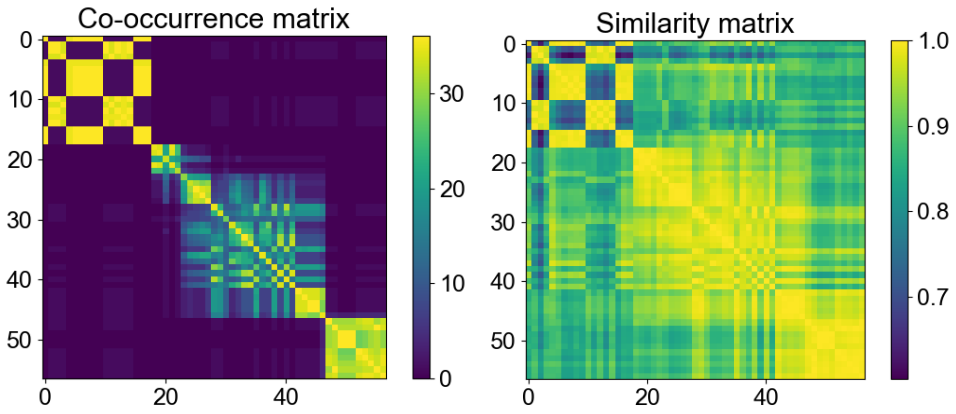
A screenshot of the stimulus presentation as was shown to participants can be seen in Figure 5.2. Each participant worked with the same set of icons but their presentation was random order.

Figure 5.2: Stimulus presentation for Test 1

5.2.4 Results

To see how users agree on the clustering, we calculated a co-occurrence matrix of the clusters made by participants and a cosine similarity matrix of the feature vectors. The resulting matrices can be seen in Figure 5.3.

We observe that the users clearly agree on the clusters. At a first glance, the co-occurrence matrix also looks rather similar to the similarity matrix. To verify that relationship, we calculated a the pairwise Pearson correlation coefficient. We find it to be 0.6, which is considered a 'moderate' linear correlation.



(a) Co-occurrence matrix of the clusters made by the users

(b) Pairwise similarity matrix of the feature vectors.

Figure 5.3: Co-occurrence matrix of the clusters made by participants and a cosine similarity matrix of the feature vectors.

5.2.5 Interpretation

The correlation matrix shows a notable level of agreement among users. We note that the 0.6 correlation value we have reported might be a conservative estimate of this agreement. This is because the comparison between the correlation and the similarity matrix is not entirely fair: the similarity matrix contains more nuanced information than what participants were able to express through their selections.

Upon closer examination, we observe that certain clusters in the similarity matrix exhibit a higher degree of ambiguity compared to others. In contrast, on some of the icons users seem to almost completely agree, as for example the first 17 items in the co-occurrence matrix. However, there are regions within the matrix where users seem to diverge in their opinions, particularly in the middle section (approximately entries 23 to 46). It is plausible that some users have grouped their selections coarsely, while others have adopted a more fine-grained approach.

When designing this test, we had hoped for a direct comparison of our method to Kolhoff et al. [2008]. The original authors kindly provided us with the playlist of songs used. However, somewhere in the past 17 years, the resulting clusters had gone missing. Many of the songs they had used were unavailable through the Spotify API, and a few songs could not be retrieved at all. Consequently, despite their enthusiastic cooperation, replicating their experiment or conducting a comprehensive comparison proved challenging. Fortunately, we were make, in part, a comparison for the second part of this test, as detailed in Section 5.3.

5.3 Test 2: Outlier Detection

This is the second test that was performed in the user study by Kolhoff et al. [2008]. This test builds upon Test 1 (see Section 5.2), by using the cluster data the participants provided themselves. It is in essence a 5-alternative forced-choice task [Cunningham and Wallraven, 2011].

5.3.1 Goal

The goal of this test is to see how well the icons represent similar music and if the clustering allows users to spot outlier songs easily.

5.3.2 Description

This task builds upon the clusters made by the participants in the task described in Section 5.2. For each user, we selected four songs from one cluster at random and add one song from another cluster at random and offer these songs to the user in a random sequence. Then asked the participant to spot the song that sounds different from the others. We repeated this 3 times for each participant.

5.3.3 Stimulus selection and presentation

We worked with the cluster data the participant had created in Test 1 (see Section 5.2). We selected all clusters that consisted of four or more songs. Then for each repetition, we randomly selected four songs from one of these clusters, and a song from any of the clusters, which we expected to be selected as the outlier. A screenshot of the stimulus presentation as was shown to participants can be seen in Figure 5.4.

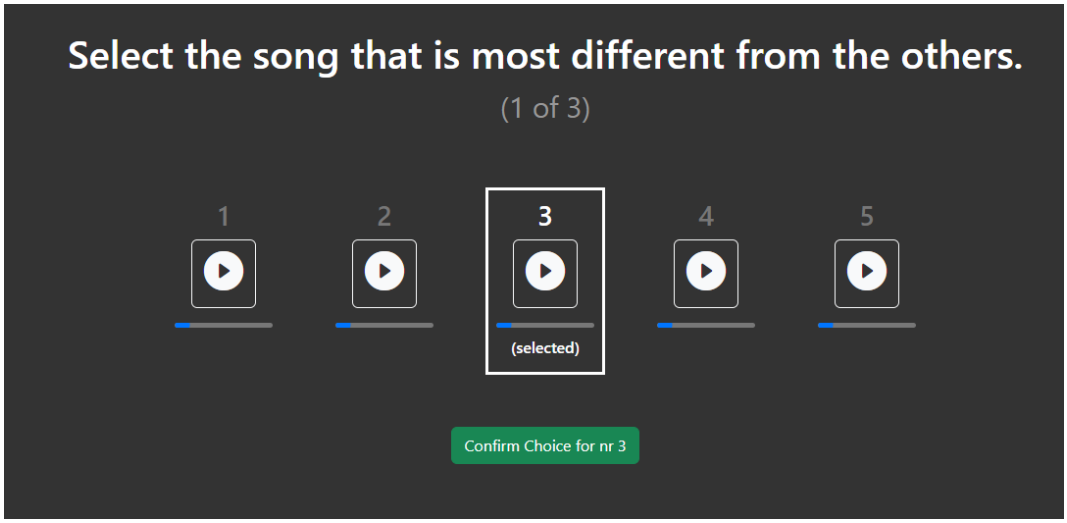


Figure 5.4: Stimulus presentation for Test 2

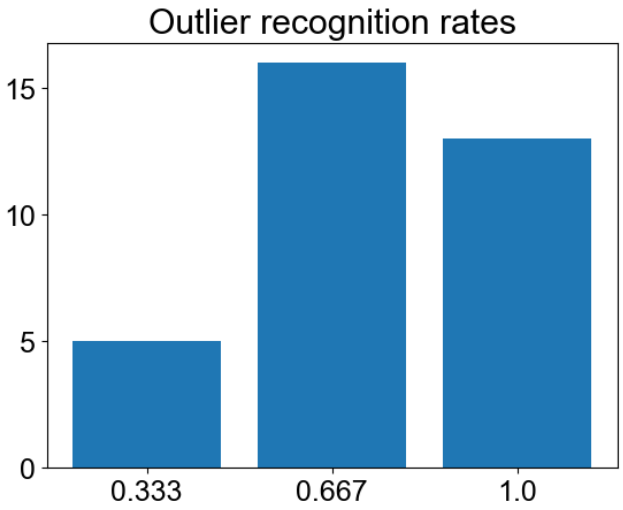


Figure 5.5: Recognition rates obtained for outlier detection in Test 2

Mean	Median	Mode	Std	Variance
0.7451	0.66667	0.66667	0.2295	0.05267

Table 5.1: Descriptive statistics of the data as displayed in Figure 5.5.

5.3.4 Results

The outlier recognition rates achieved by the participants can be seen in Figure 5.5 and the descriptive statistics of the results can be found in Table 5.1.

The expected recognition rate when of random guessing would be 0.2. It seems rather likely that our results with a mean recognition rate of 0.7451 is a considerable improvement. We observe that p = 0.000000000000002 (one sample one-tailed t-test), finding an effect size of 2.3751 (Cohen's d).

5.3.5 Interpretation

With this test, we are able to directly compare our results with the recognition rates reported by Kolhoff et al. [2008]. They reported a mean recognition rate of 0.695 with a standard deviation of 0.171. In a direct comparison of their results with ours, we used an independent one-tailed t-test, and found a small effect size 0.220 (Cohen's d), but fail to establish statistical significance for p = 0.310.

It is worth noting that their study included only 6 participants, while our study had a sample size approximately six times larger, which means that the comparison between the two studies is somewhat skewed. To gain further insights into the significance of our comparison, we estimated the posterior power of both results using an online calculator [Kane, 2018]. The results indicate that our test has 100% power, while Kolhoff et al. [2008] has 81.1% power, suggesting that both studies are likely to detect the indicated effect (although our setup is more likely to do so). However, when it comes to comparing the samples directly, the power is only 9.3%, which can be attributed to the substantial difference in sample sizes. It is important to consider these factors when interpreting and comparing the results of the two studies.

We can confidently conclude that there is a strong relationship between the clusters created by participants and the songs they represent, as evidenced by participants' ability to detect approximately 75% of outliers. Although we did not observe a significant improvement upon Kolhoff et al. [2008], our method performs at least as well, and we believe our evaluation is more robust, has higher statistical power, and is easier to reproduce. Therefore, we can assert that our results provide a solid basis for evaluating the effectiveness of our approach in comparison to previous work.

5.4 Test 3: Generalisation, Contrast and Colour Blindness Robustness

This test is set up as a matching-to-sample task in the same manner as Fuchs et al. [2014]. Note that this is a specific version of the n-alternative forced-choice task (n=8) [Cunningham

and Wallraven, 2011].

5.4.1 Goal

The goal of this test was threefold:

1. Evaluate if the 'most similar' icon aligns with the data point having the highest cosine similarity. In other words: if the icon is generally meaningful for representing high-dimensional data.

2. Evaluate if the contrast enhanced version of the icon improves performance in terms of time-on-task and accuracy for finding the most similar icon.

3. Evaluate the robustness of the icon design against colour blindness by testing the time-on-task and accuracy with a colour blind-simulated version of the icon.

5.4.2 Description

We presented the user with 9 icons that are all rather similar. One of the icons was the target icon. We asked participants to select the icon most similar to the target icon.

We performed this test for three different 'rendering modes':

- 'default', as the icon was designed and explained in Chapter 3

- 'contrast', with 100% increase of contrast between the 9 icons, as explained in Section 3.6.2

- 'colour blind', with colour blindness simulated on the colour rendering, more specifically deuteranomaly - the most common form of colour blindness

We repeated the task 9 times for each participant: 3 times for each rendering mode.

5.4.3 Stimulus selection and presentation

For this test, we wanted icons that were rather similar. Therefore, we sampled randomly from the clusters mentioned in Section 5.1.2 for k = 40. We sampled 9 clusters for 9 data points, in total 81. Among the 9 samples that were sampled from each cluster, we selected the two icons with the closest cosine similarity, randomly selected one as target and the other one as 'match'.

As for the presentation, the target icon was placed in the centre and the remaining 8 icons were arranged around the target, such that the distance between the target and the other samples is equal [Fuchs et al., 2014]. A screenshot of the stimulus presentation as was shown to participants can be seen in Figure 5.6.

To avoid order effects, we randomised the order in which the icons were presented on screen, the order in which the different rendering modes were applied and the order in which the different sets were presented.

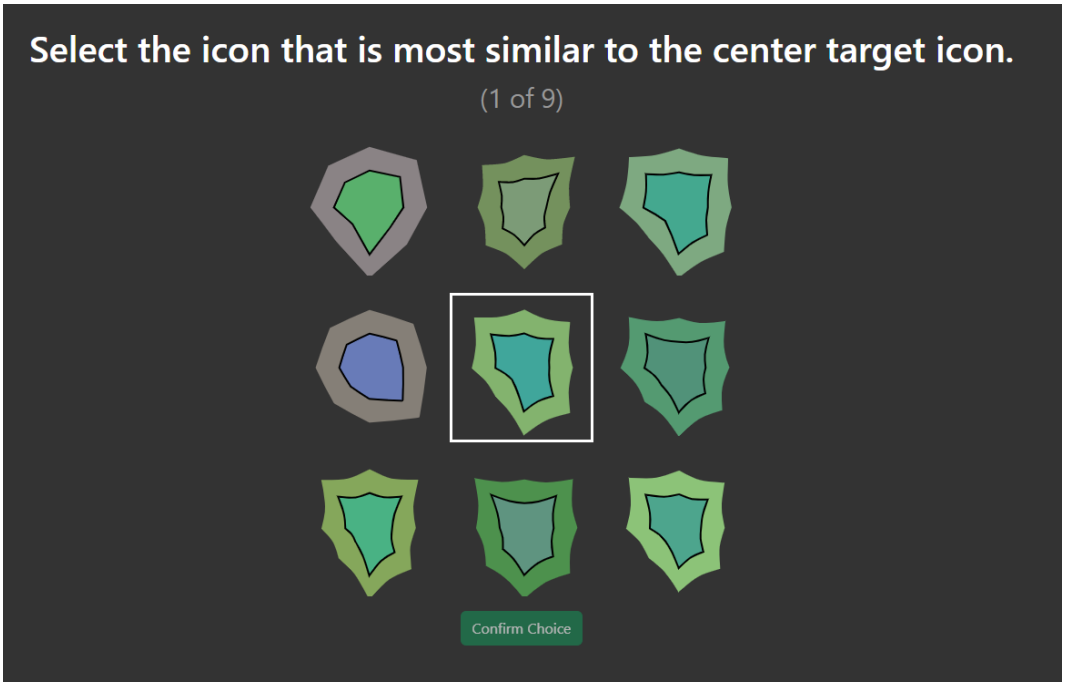


Figure 5.6: Stimulus presentation for Test 3

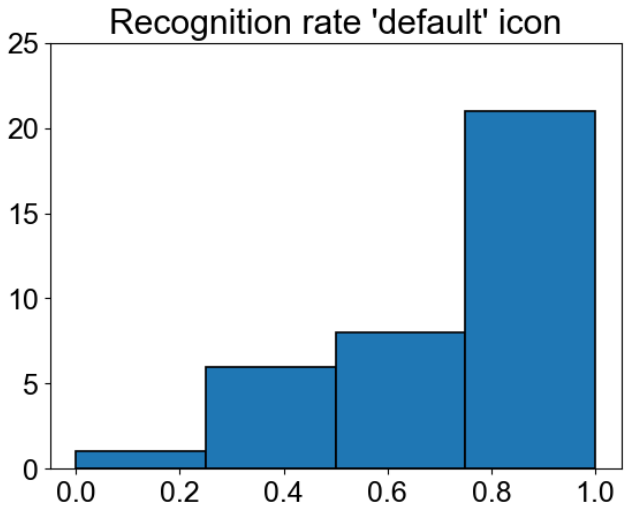


Figure 5.7: Recognition rates obtained for matching-to-sample with our icon Test 3.

Mean	Median	Mode	Std	Variance
0.70602	0.75	1.0	0.27672	0.07658

Table 5.2: Descriptive statistics of the data displayed in Figure 5.7.

5.4.4 Results

We will discuss the results for each of the goals we set out in Section 5.4.1: an evaluation of the icon as a means of representing data, an evaluation of the high-contrast version and the performance of the icon under colour blindness.

Recognition rate 'default' icon

Figure 5.7 shows the recognition rates for our icon, and Table 5.2 provides the descriptive statistics for the data. The expected recognition rate for random guessing would be 0.125. It seems rather likely that our results with an mean recognition rates of 0.70602 is a considerable improvement, allowing the user to find the icon that represents the closest point in the 8D space. To confirm we do a one sample one-tailed t-test and find that p is effectively 0 and the effect size 2.0996 (Cohen's d).

High Contrast

We are foremost interested in a comparison of the high-contrast version of the icon with the 'default' icon, to see if the contrast mode increases the performance of the icon in terms of recognition rate and time-on-task.

Figure 5.8 and Table 5.3 present a comparison of recognition rates between the high contrast and 'default' rendering of the icon. Similarly, Figure 5.9 and Table 5.4 display a comparison of time-on-task between the high contrast and 'default' rendering of the icon.

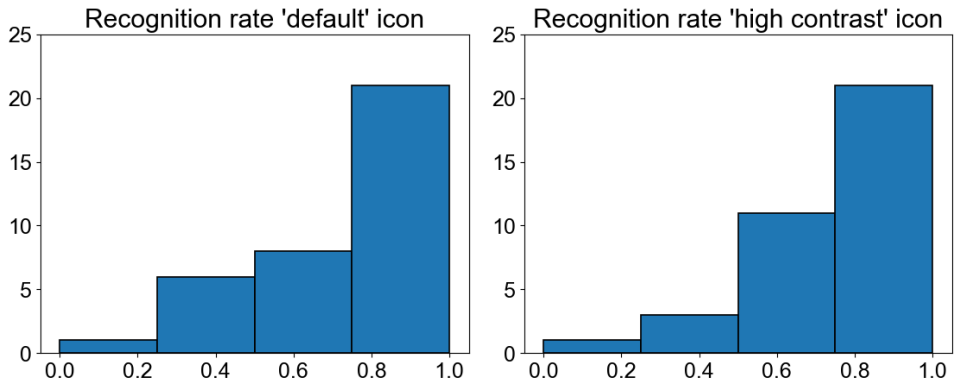


Figure 5.8: Comparison of recognition rates obtained for matching-to-sample with the 'default' and 'contrast' version of the icon.

Rendering mode	Mean	Median	Mode	Std	Variance
Default	0.70602	0.75	1.0	0.27672	0.07658
Contrast	0.78472	1.0	1.0	0.27101	0.07345

Table 5.3: Descriptive statistics of the data as displayed in Figure 5.8.

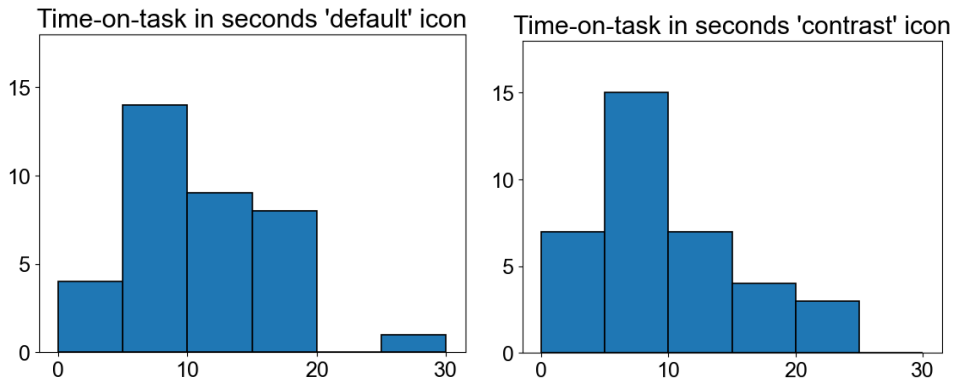


Figure 5.9: Comparison of the time-on-task for matching-to-sample with the 'default' and 'contrast' version of the icon.

Rendering mode	Mean	Median	Mode	Std
Default	10.8617	9.7415	5.44593	29.65817
Contrast	9.75386	7.812	5.38508	28.99909

Table 5.4: Descriptive statistics of the data as displayed in Figure 5.9.

We had expected the high contrast version to yield higher recognition rates than the default version. Inspecting the plots and the statistical descriptions, this seems to be the case: From Table 5.3, we notice that for high-contrast rendering, the mean recognition rate is higher than for the 'default' version of icon and indeed we see the distribution in Figure 5.8 shift a bit to the right. However, for a statistical analysis, we find a medium-sized effect size of 0.283 (Cohen's d), meaning that but fail to determine statistical significance ($p = 0.103$ for a one-tailed paired-samples t-test).

In terms of time-on-task, we had expected the high contrast icon to allow for faster selection than the default icon and that seems to be indeed the case. As with the recognition rates, we find a medium-sized effect of 0.202 (Cohen's d) but fail to establish strong significance: $p=0.084$ (one-tailed paired-samples t-test).

Colour Blindness

We are interested in a comparison of a colour-blind simulated version of the icon with the 'default' icon, to see if the redundant encoding in our icon indeed makes the icon more robust to colour blindness. Therefor we compare the recognition rates we found in the sample matching for the 'default' and 'colour blind' rendering modes. In Figure 5.10, we can compare the distribution of the recognition rates participants achieved for the high contrast version of the icon with the default rendering of the icon, Table 5.5 provides the descriptive statistics for the data.

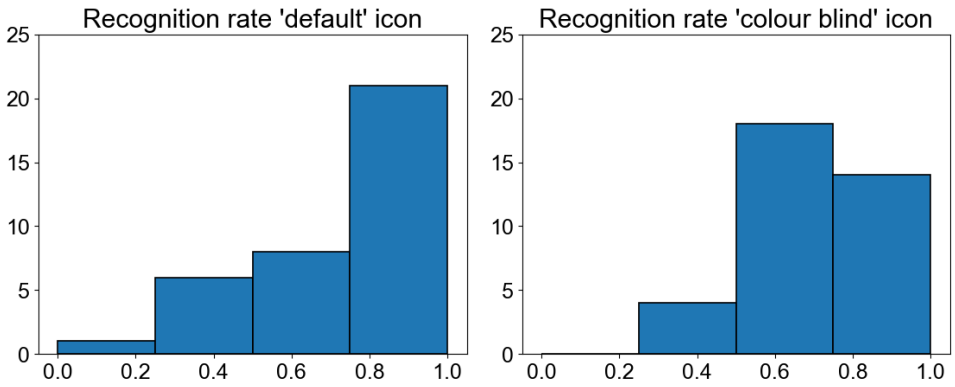


Figure 5.10: Comparison of recognition rates obtained for matching-to-sample with the 'default' and 'colour blind' version of the icon.

Rendering mode	Mean	Median	Mode	Std	Variance
Default	0.70602	0.75	1.0	0.27672	0.07658
Colour blind	0.74074	0.66667	0.66667	0.23055	0.05316

Table 5.5: Descriptive statistics of the data as displayed in Figure 5.8.

Up front, we hypothesised that the colour-blind version would under-perform slightly in comparison with the default version of the icon. There seems to be a change in the distribution, where the median value does shift from 0.75 to 0.66667 (Table 5.5. However, much to our surprise, we find that the mean recognition is a bit higher.

But we can (once again) not confirm any statistical significance between these distributions: a two-tailed paired-samples t-test yields a p value of 0.7146.

5.4.5 Interpretation

Overall, we find that the icon performs well in the matching-to-sample task. We suspect that our icon might generalise well to various types of high-dimensional data. To further investigate this hypothesis, we conducted a comparison with the star glyph experiments conducted in Fuchs et al. [2014]. In their study, they extensively tested various variants of the star glyph using a matching-to-sample approach. They assessed the impact of the number of dimensions and found that the recognition rate for the 10-dimensional glyph was, on average, 0.406, while for the 3-dimensional glyph it was 0.767. It is important to note that the data points used in their study were much more divergent than the ones in our study. Therefore, drawing strong conclusions is challenging. Nevertheless, we are pleased that our method demonstrates a high recognition rate on the matching-to-sample task with a high-dimensional icon, when compared to other research findings.

As for the high contrast icon, we speculate that in some cases the contrast might have been too strong. The contrast calculation was calculated by min-max scaling the 8D features of only 9 samples. It is possible that this process resulted in all 9 icons becoming so distinct

from each other that it became challenging to identify any similarity. It would be worthwhile to conduct further experiments using different levels of contrast, such as 50% or 75%, to explore the impact on icon perception and recognition.

As for the colour blind approach: in our analysis, we found no significant differences between the 'default' rendering of the icon and the colour-blind version. Additionally, we conducted a one-way ANOVA test on all three rendering modes, which resulted in a p-value of 0.450. This suggests that the different rendering modes do not have a substantial effect on the matching-to-sample task, indicating that each rendering mode of the icon performs similarly well. Based on these findings, we conclude that our icon is robust to colour blindness, and the redundant encoding approach we employed is effective in facilitating recognition and matching of samples.

5.5 Test 4: Search-by-icon

This test is comprised of a 'real-world task', a high-level task that a participant might encounter outside of a lab study. Most high-level tasks can be dissected such that we find the basic tasks as described in Cunningham and Wallraven [2011], which facilitates an easier analysis of the results.

5.5.1 Goal

The goal here was to evaluate how meaningful our 'search-by-icon' approach is for users. We distinguish three sub-goals in this test:

- Evaluate how close the user can imitate an icon.
- Evaluate how well the user can retrieve 'similar' music with this tool.
- Evaluate the willingness of users to adopt this tool.

5.5.2 Description

The user was presented a target song with custom icon was presented, and the search-by-icon interface (as introduced in Section 3.6.1). The participant was asked to use the tool to find the 3 most similar songs to the target song. At the end of the test, the user was asked to fill in the System Usability Scale [Brooke et al., 1996] on their experience with this interface. The System Usability Scale, or SUS, is a popular tool in usability research, which allows researchers to get an estimate of the usability of their system with a concise 10-question survey.

We identify in this real-world task two basic tasks:

- A very peculiar variant of the n-alternative forced-choice task (n = 9999).
- The task to select the parameter settings, which can be seen as an 8-fold matching-to-sample task.

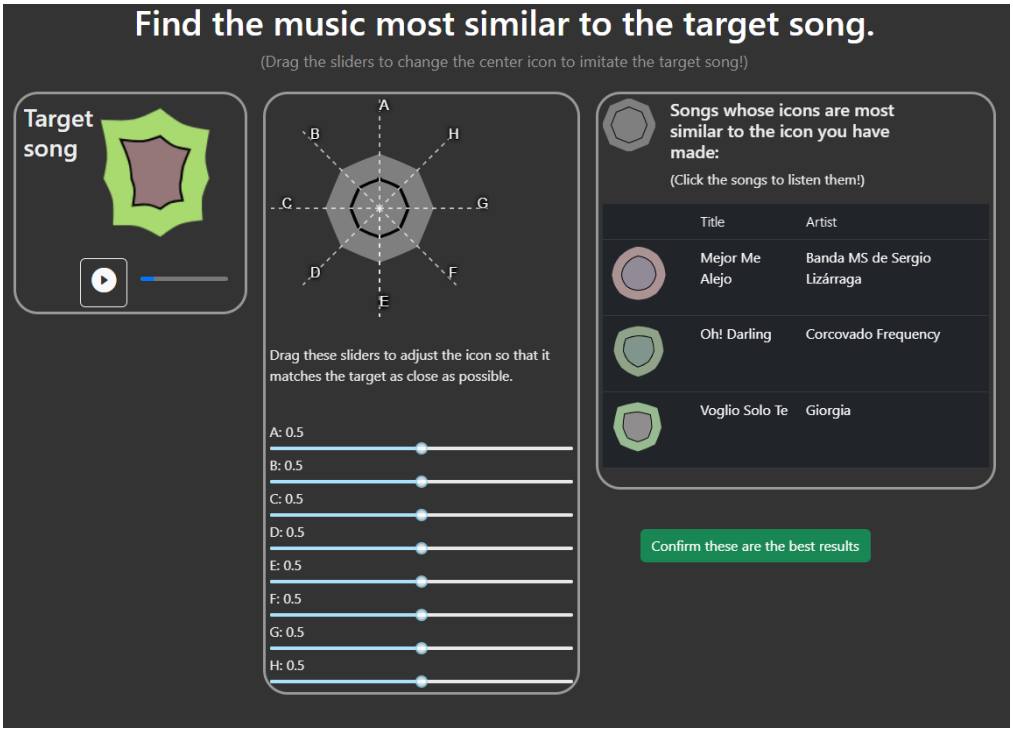


Figure 5.11: Stimulus presentation for Test 4

5.5.3 Stimulus selection and presentation

For the stimuli, we sampled randomly from the clusters mentioned in Section 5.1.2. To ensure participants were not testing the same icon, we obtained 10 icons by sampling 1 random sample from each of the clusters that were found for k = 10. This yielded us 10 distinctive data points.

A screenshot of the stimulus presentation as was shown to participants can be seen in Figure 5.11. Participants were presented with a randomly chosen icon from the selected 10 datapoints.

5.5.4 Results

We evaluate this novel method by evaluating the different sub-tasks we distinguished: how close the user can imitate an icon, how well the user can retrieve 'similar' music with this tool, and the willingness of users to adopt this tool with the System Usability Scale.

Imitation of icon

The cosine similarities between the vector of the icon as imitated by the user and vector the target icon are presented in Figure 5.12, while Table 5.6 provides the descriptive statistics related to those rates. We observe a high mean (0.98885) and median (0.9929) and conclude

that almost all users manage to create an icon with a vector that has a cosine similarity with the target icon vector over 0.975.

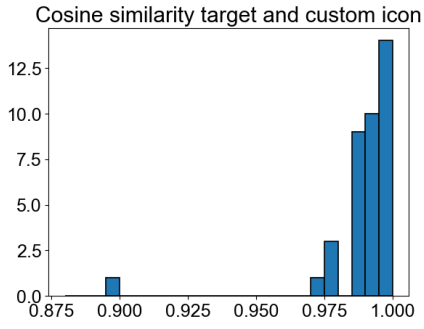


Figure 5.12: Cosine similarities between the vector of the icon as imitated by the user and the target icon.

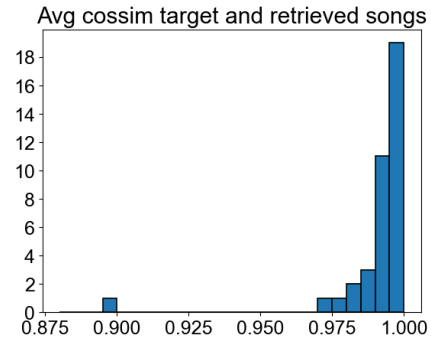


Figure 5.13: Average cosine similarities between the vector of represented by the target icon and the top 3 selected songs.

Mean	Median	Std	Variance
0.98885	0.9929	0.01676	0.00028

Table 5.6: Descriptive statistics of the data as displayed in Figure 5.12.

Mean	Median	Std	Variance
0.99087	0.99507	0.01684	0.00028

Table 5.7: Descriptive statistics of the data as displayed in Figure 5.13.

Retrieved songs

Average cosine similarities between the vector of represented by the target icon and the top 3 selected songs can be found in Figure 5.13, the accompanying descriptive statistics can be found in Table 5.7. We find that these results reflect the close imitations that were achieved of the target icons.

System Usability Scale (SUS)

The SUS consists of the ten questions:

1. I think I would like to use this product frequently.
2. I found it unnecessarily complicated.
3. I found the product easy to use.
4. I think I need technical support to use the product.

5. I found the different functions of the product well integrated with each other.
6. I felt there were too many contradictions in the product.
7. I can imagine that most people can quickly get to grips with the product.
8. I found the product cumbersome to use.
9. I felt confident while using the product.
10. I had to learn a lot about the product before I could use it properly.

Each of these statements was ranked with the Likert Scale anchored with 1 for 'fully disagree' and 5 for 'fully agree'. The answers that were given in response to each of the questions in the SUS can be seen in Figures E.4 and E.5.

Calculating the SUS score involves three steps:

1. Calculate the score for odd-numbered questions by subtracting 1 from the number of points earned
2. Calculating the score for even-numbered questions by subtracting the number of points earned from 5
3. Sum the scores from steps 1 and 2 and multiply the total by 2.5 to obtain the SUS score

The results can be interpreted based on the range of scores obtained, with scores above 80.3 indicating a top 10% ranking, scores above 68 indicating a top 30% ranking, and scores above 68 indicating a top 50% ranking. A score above 80.3 is interpreted as 'excellent', scores between 68 and 80.3 as 'good', scores between 50 and 68 'mediocre', and anything below 50 'bad', as illustrated in Figure 5.14.

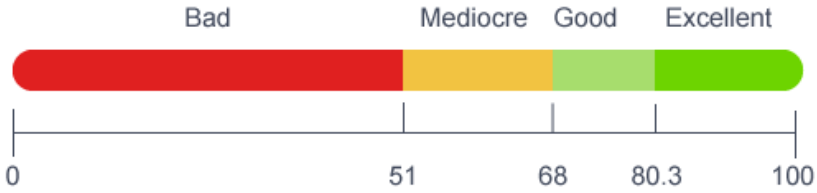


Figure 5.14: An interpretation of the SUS score ranges.

In Figure 5.15, we can see the resulting SUS scores from the participants. Mapping these results as per Figure 5.14, we find the distribution in Figure 5.16, where we count 13 'bad' results, 7 'mediocre', 11 'good' and 6 'excellent'. We realise that flattening the user experience into such a score is a gross simplification. Nevertheless are excited to see that 25 of 38 participants seem willing to accept our model.

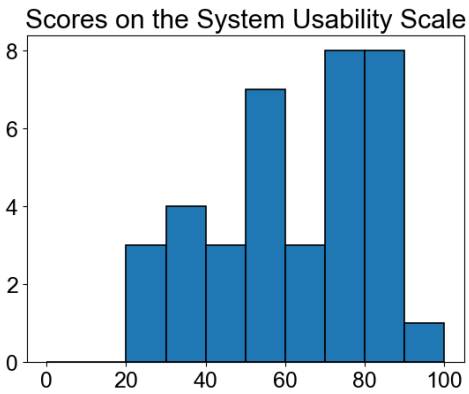


Figure 5.15: SUS Scores for the UI used in Test 4.

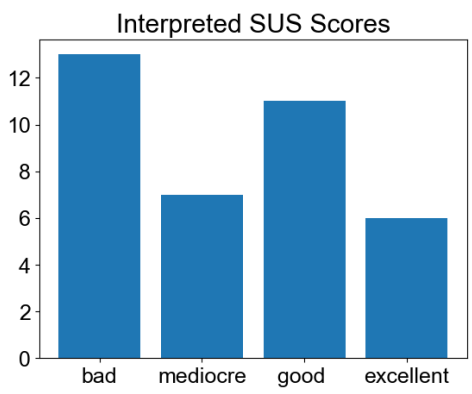


Figure 5.16: Interpretation of the SUS Scores for the UI used in Test 4.

5.5.5 Interpretation

We find that users were able to work with our tool at a much higher level of proficiency than we had anticipated. The positive response from participants, as indicated by the SUS scores, exceeded our expectations, finding more participants willing to accept than reject it. While it is important to note that the SUS is an oversimplified measure of user experience, this result indicates a promising potential of our tool.

We believe that this tool has interesting potential for user-guided search and exploration, particularly in scenarios where users have an open mindset and aim to discover music similar to a 'seed' song, as that is almost literally what this tool does.

Due to the novelty of our approach, making direct comparisons with other work challenging. We do maintain that our results bring other innovative approaches, such as the search for audio based on a mental image of the desired sound [Knees and Andersen, 2016] another step closer.

5.6 Test 5: Search-in-playlist

This test is comprised of a 'real-world task', a high-level task that a participant might encounter outside of a lab study: finding music in a playlist that is similar to a target song. In our approach, we mainly focus on the time-on-task and the number of steps (number of songs played) taken to complete the task.

5.6.1 Goal

The goal of this test was to evaluate the effectiveness of the icon and its sorting properties in a playlist-setting, and compare it to what is currently most common in streaming services: a presentation with album art.

5.6.2 Description

Participants were asked to select their top 3 songs from playlists with both album art as our custom design. Both were repeated one time.

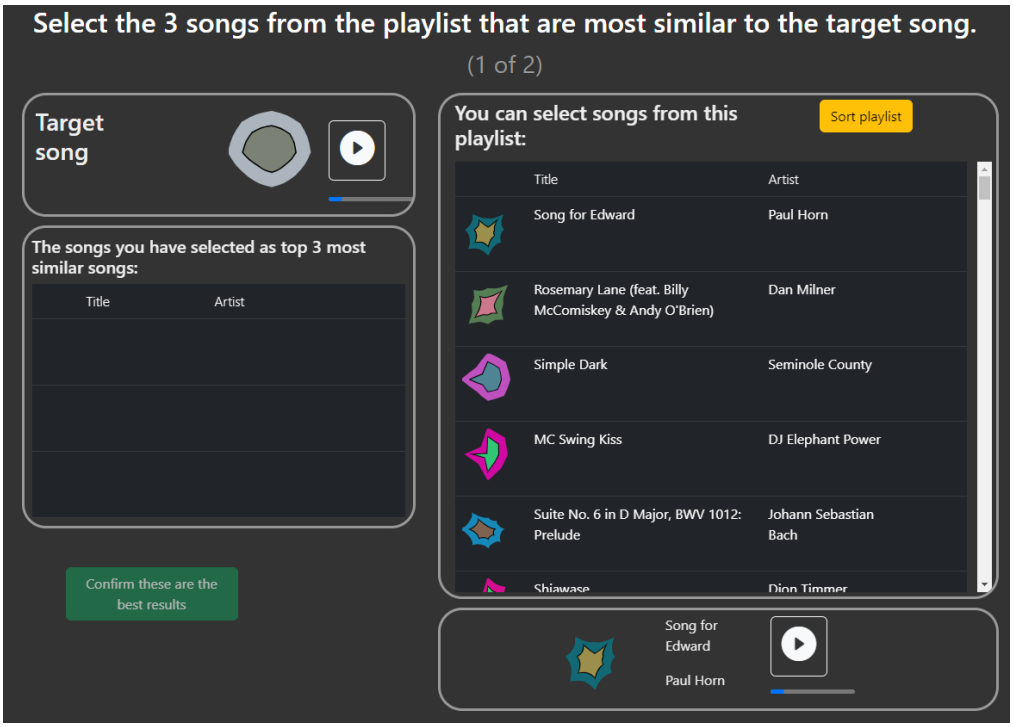


Figure 5.17: Stimulus presentation for Test 5

5.6.3 Stimulus selection and presentation

For the stimuli, we sampled randomly from the clusters mentioned in Section 5.1.2. To ensure that there is a diversity in the selection yet still the possibility to make clusters, we sampled 10 datapoints from each of the 10 clusters that were found for k = 10. From these data points, we made a playlist of the resulting 100 datapoints. With this process we created two playlists. The target songs were also obtained by sampling from the k = 10 clusters: from each of the 10 clusters we randomly sampled 1 song, yielding 10 possible 'target' songs.

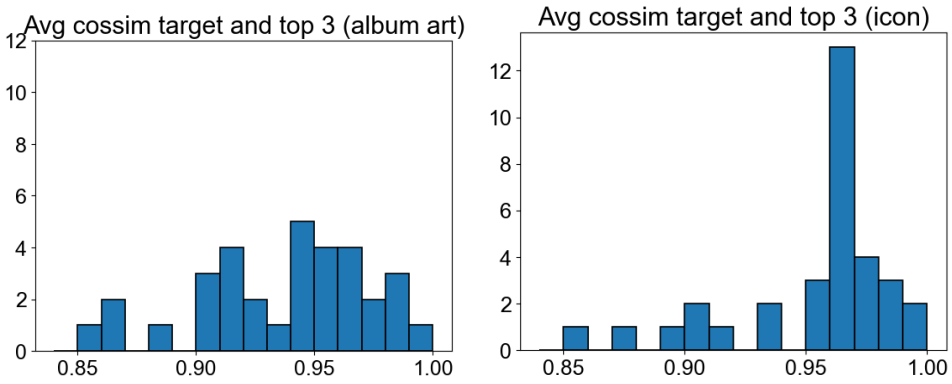
A screenshot of the stimulus presentation as was shown to participants can be seen in Figure 5.17. To prevent order effect, the target song was selected randomly from the selection of possible target songs, the playlist order was randomised for each participant, as was the order in which they were presented with custom icon and album art icons.

5.6.4 Results

We evaluate the similarity of the top 3 compared to the target vector, the Time-on-task and plays per task, as well as a couple of open questions.

Retrieved songs

Average cosine similarities between the vector of represented by the target icon and the top 3 selected songs can be found in Figure 5.18, the accompanying descriptive statistics can be found in Table 5.8. We find that both methods span the same range of cosine similarities but that the icon method enables users to retrieve songs with slightly higher similarity (one-tailed paired-samples t-test: p = 0.03, Cohen's d: 0.4688). This seems to make sense as this is the similarity the icon is oriented towards.



(a) Album art icons (b) Custom icon

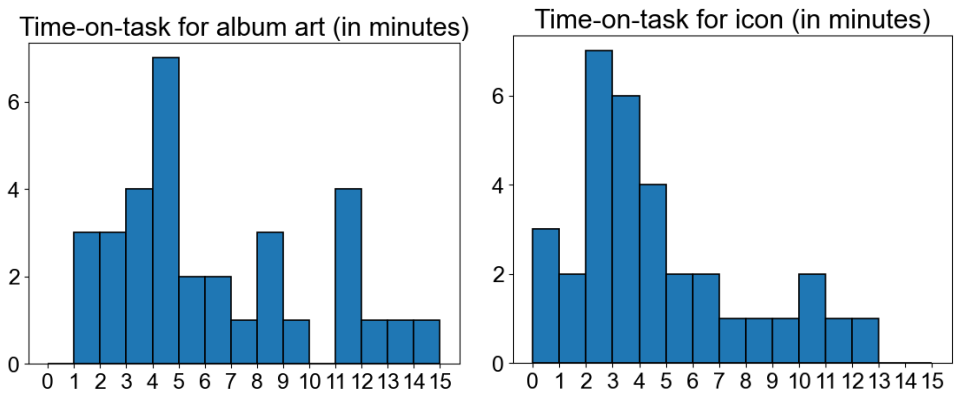
Figure 5.18: Average cosine similarities between the vector of represented by the target icon and the top 3 selected songs as obtained in Test 5.

Icon	Mean	Median	Std	Variance	Min	Max
Custom	0.93805	0.94513	0.03614	0.00131	0.85134	0.99336
Album Art	0.95459	0.96721	0.03327	0.00111	0.85754	0.99093

Table 5.8: Descriptive statistics of the data as displayed in Figure 5.18.

Time-on-task

The time-on-task per participant for both album art and custom icon can be found in Figure 5.19, with descriptive statistics in Table 5.9. We observe a strong speedup of the mean time-on-task: well over a minute. We confirm the results with a left-tailed paired t-test and find p: 0.00201 and effect size: 0.47292 (Cohen's d).



(a) Time-on-task when using album art icons (b) Time-on-task when using the custom icon

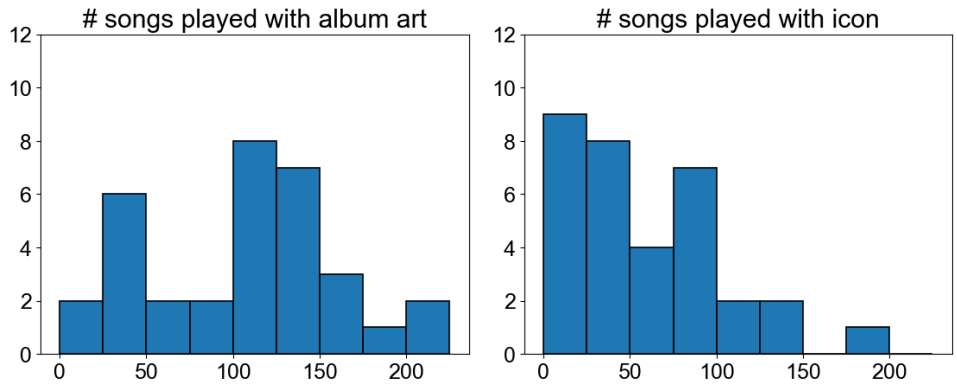
Figure 5.19: Time-on-task for album art and custom icons respectively.

Icon	Mean	Median	Std	Min	Max
Album Art	6:25	4:49	3:45	1:08	14:12
Custom	4:44	3:54	3:11	0:39	12:14

Table 5.9: Descriptive statistics of the data as displayed in Figure 5.19, formatted as mm:ss.

Plays per task

The plays per task per participant for both album art and the custom icon can be seen in Figure 5.20, with descriptive statistics in Table 5.10. We observe that the ranges of both are about the same but observe a much smaller mean and median number of plays when using the custom icon. A paired t-test confirms, p: 0.00001, effect size: 0.93089 (Cohen's d).



(a) Songs played per task when using album art icons (b) Songs played per task when using the custom icon

Figure 5.20: Songs played per task for album art and custom icons respectively.

Icon variant	Mean	Median	Std	Variance	Min	Max
Album Art	103.912	113.5	55.867	3121.139	10	222
Custom	57.735	40.0	42.389	1796.842	7	192

Table 5.10: Descriptive statistics of the data as displayed in Figure 5.20.

5.6.5 Open Questions

At the end of the study, participants were asked three open questions about their experience with the playlist task. We summarise and highlight the responses here.

In addition, have incorporated the feedback from the participants into our suggestions and recommendations for future work, as outlined in Chapter 7.

Q1: How did you experience using the custom icon differ from the regular setting?

To this first, question 23 participants mentioned a positive experience. 14 mentioned explicitly that they experienced that it made their task of music selection easier. 3 participants mentioned that they considered the album art icon more fit for the task, of which 2 explicitly mentioned that they found the album art to give more information about the song than the custom icon.

Q2: Was there anything surprising or unexpected?

To this second question, 5 participants mentioned that they were surprised how well the icon had supported their task. In contrast 5 participants mentioned they had noticed outliers, which made them doubt how well the icon really reflected the music content.

Q3: What could be done to improve the icon?

To the third and final question we got some very concrete feedback from participants: 4 participants mentioned that they would like the icon to be more expressive, in terms of shape and colour. In particular, 3 mentioned the colours as a bit bland. In addition, 12 participants expressed a longing for a better understanding of the parameters of the icons: something more semantic or a more elaborate explanation, several of them mentioned a desire for genres mapped to axis or colours.

5.6.6 Interpretation

For this task, we obtained strong quantitative results that demonstrate the effectiveness of our tool. We observed a significant speedup in the mean time-on-task, with participants completing their tasks well over a minute faster, compared to a presentation with album art. Additionally, participants listened to almost 50% fewer songs before finding a satisfactory selection.

Comparing the selections participants made when using the icon or album art, we find that when using our icon, participants selected songs with with similar or slightly higher

cosine similarity to the target song. The visual cues provided by our icon may have influenced participants' selection process, leading not only to faster decision-making, but also to different choices.

We found that many participants expressed a strong appreciation for our icon and its ability to convey similarity. However, we also acknowledge the viewpoint of some participants who preferred album art icons, as they provide contextual information that allows users to infer cultural details about the music. While we recognise the value of album art in certain contexts, we maintain that our icon offers a valuable improvement, especially considering the significant variation that can exist among songs within an album.

Chapter 6

Discussion

In this chapter, we provide a discussion of the interpretation of the results obtained from our method. We delve into the visual clustering properties, generalisation capabilities, robustness to colour blindness, contrast enhancement features, the application in playlists and the search-by-icon functionality. Then, we compare our work with our main inspiration Kolhoff et al. [2008] and finally, we critically reflect upon the broader implications of this work, considering its potential impact and significance beyond the immediate scope of our study.

6.1 Visual clustering

We are pleased with the visualisation results, particularly the robust clustering capabilities, as qualitatively assessed in Chapter 4 and quantitatively evaluated in Chapter 5. With a free-grouping task (as described in Section 5.2), we found that users strongly agree on the visual clusters, and the clusters created by users are linearly correlated to the cosine similarity of the underlying feature representations.

However, we observed in Section 4.2 that our results strongly differentiate between classical music containing violins and classical music with piano. The CLMR model we employ for feature extraction is a CNN, a type of network that tends to have a bias towards texture/timbre. We suspect that our model also exhibits a predisposition towards timbre, which is reflected in the clustering outcomes. This could also explain the outlier mentioned in Section 4.3, where a song with a visually distinct icon seemed to fit well with the EDM genre of its surrounding icons. However, the song featured distinctly different instruments such as a guitar. It is plausible that such specific sensitivities of the model are influenced by the downstream task for which it was trained. It might be worthwhile to explore the visual clusters that our method could produce by employing a different model.

In addition, we noted that due to the stochastic nature of the UMAP algorithm, our dimensionality reduction leads to, in part, an arbitrary mapping. Employing a different random seed would have resulted in completely different icons for the same music, and possibly a report with entirely distinct figures. However, as demonstrated in the convergence experiments outlined in Section 3.4, the UMAP algorithm effectively preserves the

(dis)similarities between the data points. Therefore, we assert that while a different initialisation would have generated different icons, the findings from our tests provide valuable insights not only for the specific random seed *1989* but also for our overall methodology.

6.2 Generalisation

We have quantitatively investigated the possibilities of using the icon in a more generic sense as a representation of any high dimensional data for a matching-to-sample task, as described in Section 5.4. Our findings indicate that the icon enabled participants to successfully identify close points within high-dimensional space.

We hypothesise that our icon has useful properties that may generalise for visual search and comparison tasks involving diverse data sets with diffuse class boundaries, much like our music dataset. To validate this assertion, a comparison with other glyphs is necessary. It is important to note that visual search and similarity identification are tasks that are distinctly different from the reading of values from a datapoint, for which sometimes glyphs are also employed. We think that our icon is not well-suited for the latter task.

In Section 5.4.4, we have already speculated on a comparison of our method with the star glyph experiments as per Fuchs et al. [2014], after which we have modelled this experiment. Our similar setup and strong results leads us to suspect that our icon performs well as a more general representation of high-dimensional data. However, we are hesitant to draw all too bold conclusions, as our study employed a different number of dimensions and the diversity of data points in their tasks was significantly greater than ours. Additionally, while we used cosine similarity to determine the closest point, Fuchs et al. [2014] employed Manhattan distance and included scaled versions of data points as stimuli. Consequently, employing our cosine similarity metric would have resulted in the selection of a different 'correct' icon. Here we are confronted once again with the observation that the variance in tasks and measures for effectiveness pose obstacles for comparing visualisation approaches [Gleicher, 2017].

6.3 Colour Blindness Robustness

We have quantitatively investigated the possibilities of generalisation of the icon and its robustness to colour blindness in our user study with an matching-to-sample task, as described in Section 5.4

We also had participants do the same matching-to-sample task with simulated deuteranopia, the most common form of colour blindness. While such simulated distortion of the colours does seem to influence the performance of the participants a bit, we find that there is no statistically significant deterioration in performance. This observation suggests robustness to colour vision deficiencies, which we attribute to the redundant encoding of variables within the icon.

While colour blindness is mentioned in the more recent literature on the use of colour and visualisation, we have not encountered work in our literature review on glyphs and music visualisation that specifically tested their icon for robustness against colour blindness

or even made recommendations for this. We are thus not aware of other work that proposed a glyph design incorporated colour yet is proven to be robust to colour blindness.

6.4 Contrast enhancement

The first visualisation results of our contrast enhancement approach seemed promising in Section 4.6. However, we were unable to validate its effectiveness quantitatively in Section 5.4.4. It seems premature to conclude that this feature does not work at all.

We think that this feature is most suitable for detecting similarities in icons that are relatively similar, similar to our matching-to-sample task. In our test setup, however, the contrast may have been too high. The contrast enhancement is based on min-max scaling of a subset. The subset contained in this case only 9 datapoints. It is plausible that all 9 icons became so distinct that is was challenging to identify any similarity. In playlists with already high diversity, this does not happen. There, in fact, the contrast enhancement sometimes makes little difference (depending on the diversity).

There may very well exist an optimal range of contrast, which likely depends on the diversity of datapoints as well as the number of datapoints. Finding this range requires additional experiments to determine the parameters for the right amount of contrast for a given level of diversity and number of datapoints.

6.5 Use in playlist

In Section 5.6, we evaluated the use of the icon in a real-world playlist setting. Participants were asked to select the songs from a playlist that were most similar to a target song, once using our icon and once with album art icons, which is a presentation commonly employed by streaming services. Assessing the performance of participants on this high-level task, we found that using the icons largely and significantly decreases both time-one-task and the amount of steps required to find similar music in a playlist. Compared to a presentation with album art, we achieved a time savings of over a minute and nearly 50% fewer songs listened to before participants were satisfied with their selection. These results present a direct answer to our main research question.

When presented with album art, participants listened to each song in the playlist before being confident in their selection, using our icon they listened to only half of them. This is an indication they trusted our design. Participants confirmed their trust in the icon when filling in the questionnaire afterwards, where 14 participants mentioned spontaneously that they experienced the icon had made their task of music selection easier and 5 participants explicitly mentioned that they were surprised how well the icon had supported their task.

To asses whether that trust was rightful, we compared the songs participants had selected as 'most similar' to the target song for the album art presentation and our icon design. When presented with our icon, users selected songs whose latent vectors exhibited similar or slightly higher cosine similarity to the latent vector of the target song than they did when presented with album art. As our icon provides visual cues based on these latent vectors, we hypothesise that it may have influenced the selection process, resulting in users choosing

different songs compared to alternative approaches. This suggests that the icon not only facilitated faster decision-making but also guided users towards distinct song choices.

6.6 Search-by-icon

We evaluated this novel interface for music discovery through reverse search in Section 5.5 with a real-world task. We had anticipated that participants would experience difficulties in imitating an icon using this novel interface. The redundant mapping of each parameter to two visual channels made it not straightforward to understand the impact of adjusting each slider and the lack of semantic meaning in the features themselves did not seem helpful either. However, we found that users were able to create icons with parameter settings that exhibited a remarkably high cosine similarity. In addition, they successfully retrieved music with a high cosine similarity to the target song, which follows logically from the highly similar vector they reconstructed and our use of cosine similarity as search metric.

Contrary to our expectations of participants encountering difficulties and experiencing frustration, in the SUS questionnaire more than half of the respondents agreed or strongly agreed to the statement "I found the product easy to use". Aggregating the survey responses into a single SUS score, we the number of participants that indicate a 'good' to 'excellent' score outnumber the participants that did not find the product usable enough to accept it. We reiterate that while the SUS provides a clear and popular measure of user experience, it is a gross oversimplification and acknowledge that the obtained ratings by no means captures the full and nuanced user experience. However, we still embrace the results obtained from the SUS questionnaire, as a useful indicator of the overall usability of the tool that helps us see its potential and in what direction it could be developed further.

Even though this novel tool turned out to be more usable than we initially anticipated, it is important to note that participants in the user study still expressed a strong desire to understand the meaning of the icon's parameters. This is a very valid concern, as the effective use of the icon, both in the context of this tool and in playlists, relies on an some understanding of the icon. At the bare minimum the use of the icon relies on a learned interpretation, and in a more ideal case, an understanding of the parameters. Introducing parameters with more semantic meaning would facilitate such understanding, potentially enhancing the usability of the tool and benefiting a larger user base. Suggestions on how to achieve this are made in Section 7.3. The search-by-icon interface itself is well-suited as a tool for learning the mapping between the parameters and their visual representation.

6.7 Comparison with Kolhoff et al. [2008]

We based this project on the pioneering work of Kolhoff et al. [2008], which introduced the concept of creating custom content-based music icons. For this reason, we were eager to compare our work with the study conducted by Kolhoff et al. [2008]. We attempted to replicate one of their experiments, which consisted of two consecutive tasks. We encountered difficulties in reproducing the first task due to the unavailability of the music they used (as described in Section 5.2). The second part of the experiment involved a 5-alternative

forced-choice task (as described in Section 5.3). Initially, we expected comparison to this task to be very feasible as Kolhoff et al. [2008] provided a clear metric: recognition rates for auditive outlier detection. We were able to make a superficial comparison but found that power of a direct comparison was low: the wildly different sample sizes hindered a robust benchmark comparison. This highlights the complexities of visualisation research.

Despite these challenges, we are confident in making a statement that our method is at least as effective as the approach presented by Kolhoff et al. [2008]. It is important to note that our work benefits from the advancements made in the field of deep learning, which have occurred in the 17 years since the publication of Kolhoff et al. [2008]. These advancements have significantly improved the quality of available features, and we specifically leverage the advancements in the field of representation learning.

Furthermore, we believe our method has undergone more comprehensive testing and is easier to reproduce and compare. And while we built upon their concept of dual colouring, we have enhanced the method by incorporating more expressive glyphs that redundantly encode all input. This additional redundancy improves the robustness of our method, particularly in addressing colour blindness, which was a limitation of the previous approach. Moreover, we have extended the range of possible applications of the icon by introducing reverse search functionality.

6.8 Critical Reflection

In this section, we provide a critical reflection on some aspects of this project, in particular how it is susceptible to bias and vulnerable to the whims of privatised data.

6.8.1 Bias

This work contains a strong bias towards western music. That bias stems from the fact that the model we work with has been trained to on a dataset of western music, as are most MIR models. Hidden bias is a well-known problem with deep learning models. The model we employ may very well be biased in instrumental, cultural [Holzapfel et al., 2018], sexist [Shakespeare et al., 2020; Melchiorre et al., 2021] or other ways [Youngblood et al., 2021] that we are currently unaware of. This has consequences for the application of this work. Applying this visualisation without critical thought will amplify bias: it will make some artists overlooked and render some music styles invisible. Technological products establish structures that influence the way people live their lives and determine the range of social opportunities available. Who knows what masterpieces might stay hidden, how listening habits may be guided, moods manipulated and personal identities shaped differently when deploying one model versus another [Chodos et al., 2019].

6.8.2 Privatised data and reproducibility

In this work, we have objected against using private, closed source data, such as the popular and high quality Spotify API features. This was due to concerns for reproducibility, explainability and open research, amongst others.

Despite our objections, we have not been able to circumvent the use of closed-source API's: the large majority of music produced on this planet is property of one of three major record companies. To obtain music data to extract features from, we have succumbed to using the Spotify API to retrieve 30 second preview samples.

During developing this project we have seen a confirmation of our concerns. In a development project with the span of a few months, we have already struggled with the rapid changes in what songs are available from the Spotify API due to ever changing contracts with music labels. This is an ongoing problem in music research and reason why the music data that underlies benchmark datasets like the MSD, are only available through secretive and informal channels.

Chapter 7

Conclusions and future work

This chapter gives an overview of the project's contributions and draw our conclusions. Finally, some ideas for future work will be discussed.

7.1 Contributions

The main contributions of this work are: a new icon design that can be used to represent latent features of music files, and a novel interface for music discovery. In this section, we will summarise these contributions.

7.1.1 Icon

We propose a new content-based icon for music files. The icon builds conceptually on the star glyph and relies heavily on redundant encoding of latent variables. The icon is more expressive than previous work and seems robust to colour blindness. We also contribute a means of sorting the icons and the possibility of increasing contrast in a subset.

We find that users strongly agree on the visual clustering properties and that they can audibly identify songs from a different visual cluster with a mean recognition rate of 0.7451, whereas random guessing would have yielded a mean of 0.2. We interpret this as a confirmation of not only the visual clustering properties, but also an indication that the representations used capture some characteristic properties of different types of music.

The icon allows users to accurately identify 75% of close points in high dimensional space in a matching-to-sample task. Most interesting, for this task the icon does not perform significantly worse when deuteranopia colour blindness is simulated. This suggests robustness for colour vision deficiency. We suspect that our icon has useful properties for visual search and comparison tasks for other data too.

Finally, using the icon in a real-world playlist setting significantly decreases both time-one-task and the amount of steps required to find similar music in a playlist. Comparing to a presentation with album art, as is common in streaming services, we have observed a speed up of over a minute and almost 50% less songs were listened to before the participants were content with their selection.

7.1.2 Search-by-icon: a novel interface for music discovery

Our second main contribution is a new interface for search and exploring: the possibility to create an icon and retrieve music whose content is related to such an icon. Both the icon and the resulting list of most similar songs are updated real-time. We find that the icon does not have to be a perfect imitation to retrieve music that seems rather agreeable with the icon.

We found that the icons users created an icon whose parameter settings have a very high cosine similarity with the latent vector of the target song. Users also retrieved music with high cosine similarity, which logically follows from the highly similar vector they were able to reconstruct. Despite the fact that we had expected participants to experience difficulty using this novel tool, more than half of the respondents agreed or strongly agreed that they found the product easy to use.

7.2 Conclusions

We started this thesis project with the hypothesis that that visualisation of characteristics of a song can improve user-guided search and exploration by enabling visual search. From this, we formulated our main research question: **How can a visual representation help speed up identification of a song with distinctive characteristics?**

We first conducted an extensive literature review, addressing the literature related to each of the three sub-questions:

1. *What are distinctive characteristics of a song and which features can be used to represent those?* For which we conducted a literature review of the broad range of features used in Music Information Retrieval (MIR).

2. *What role can latent variables play in the representation of characteristics of a song?* For which we conducted a literature review of the work on representation learning in MIR.

3. *How can features be mapped to a visual representation?* For which we conducted a literature review on music visualisation and glyphs.

Then, to formulate an answer to the main research question, we proposed a possible solution that may help speed up identification of a song with distinctive characteristics. In the preceding chapters, we have detailed our experimental design: a new content-based icon for music songs, based on a latent feature representation, that allows for faster identification of music with similar properties. In addition, we propose a novel Search-by-icon UI that exploits the possibilities of our parameterised icon and enables a reverse search-by-image approach.

To verify to what degree our experimental design is an answer to the main research question, we have performed a user study that tested 5 different aspects of our icon and proposed interface. We found that users agree on visual clustering, that it allows for outlier detection and the identification of items of high similarity in the latent space. In addition, our tests indicate robustness to colourblindness. When searching for music similar to a

target song, we find that users are faster and have to listen to less songs to complete their selection, when presented with our icon than with a regular album art presentation. As for the Search-by-icon approach, we find that users are able to closely imitate the icon of a target songs and retrieve songs that are very close in the latent space.

We conclude that our icon design successfully represents characteristics of songs and enables the user to faster identify songs with certain properties. This new method also allows for new ways of exploring music.

7.3 Future work

In this section, we discuss potential areas for further investigation and make recommendations on how to proceed. In particular, we see fruitful directions for further improving the icon itself, as well as the UI and our evaluation methods.

7.3.1 Improvements on the icon

We see fruitful direction for further improvement in the areas that were identified in the user feedback: more expressive colours and shapes as well as more semantically meaningful features.

Expressiveness and colour

Although we consider our icon design to be much more expressive than our starting point Kolhoff et al. [2008], we see there remains room for improvement in the expressiveness of the glyph. In particular in the area of colour. Several participants of the user study mentioned that the colours were often a bit bland and it would be interesting and worthwhile to investigate how the colours can be more distinguished. One direction could be to not align more dominant colour channels with the most variant axes, another direction to investigate could be to find out how the embedding could be spread out better over the latent space, of which now entire regions remain unused. The challenge would be to also maintain clustering properties.

In addition, the shape and curvature may be improved further. Currently the 'curve-shape' parameter has no effect if the 'strength' of the curvature is zero or close to zero. Alleviating this singularity point would further benefit the expressiveness of the shape.

Semantic features

We find that users yearn for semantically meaningful parameters, as became very clear from the user study. Despite their many strengths, the latent features are too abstract for any user that is not used to working with abstract data. Perhaps they are too abstract also for users that are used to working with abstract data. In any case, a next step may be an additional mapping of the latent features to more semantically meaningful ones. A possible way to achieve this would be to add one more layer to the CLMR model and fine-tune it on a dataset that provides the parameters we would like to estimate, effectively providing

an extra (non-linear) mapping. Depending on the preferred features, such a dataset may already exist, for example or can be constructed or crowd-sourced (like Ellis et al. [2002]).

7.3.2 Improvement on the UI

We see directions for further work on the UI in particular in the novel Search-by-icon interface and in the implementation.

Search-by-icon

Dragging the parameters in the drawing of the icon itself may be a good upgrade for the Search-by-icon UI. More semantically meaningful features will directly improve the user experience in the Search-by-icon interface. We also see that despite the strengths of the redundant mapping, it may be confusing for users that one slider changes multiple properties of the icon. In the user test, one user expressed a wish for more simplicity, suggesting that perhaps less properties could be modified by the user, or less sliders used.

Implementation

The current implementation leaves plenty of room for algorithmic optimisation. However, we see the most interesting directions for this project in adding the CLMR model to run on a background server, so songs can be added to a collection dynamically. Currently, we work with a fixed dataset, but a more flexible approach would be required for this products acceptance to a larger audience.

7.3.3 Evaluation

A more qualitative evaluation, which could be in the form of interviews with participants. We expect that will yield a deeper understanding of the target users and fruitful directions for a next design iteration.

Second, we recommend more investigation of the interplay of colour and shape. In the literature review, we have seen that there is a lot to be still researched. In this work we have worked with the few guidelines available but acknowledge that a more sturdy scientific foundation can only improve our method and the work that follows. For an evaluation of these relationships we recommending further experimenting with forced-choice tasks.

Finally, it would be interesting to see how our icon performs when compared directly to a star glyph. Test 3 was set up in a fashion much like Fuchs et al. [2014], but we did not follow their experiments closely enough to confidently compare to their work. We consider testing both the star glyph and our icon with the same distribution of data and number of dimensions 'extremely interesting'.

7.3.4 Summary icons for playlists

We have proposed new icons to use in playlist, however the cover images of the playlists have remained unchanged and predominantly unrelated to the musical content they represent. Currently, the images representing personalised playlists are composed of one album

from one artist that is in the playlist, as illustrated in Figure 7.1. Exploring the possibilities of an icon that captures characteristics of the music within the playlist would be a logical next step. The objective would be to design an icon that provides a succinct visual summary of the playlist's musical characteristics, offering users a visual cue of the music contained within.

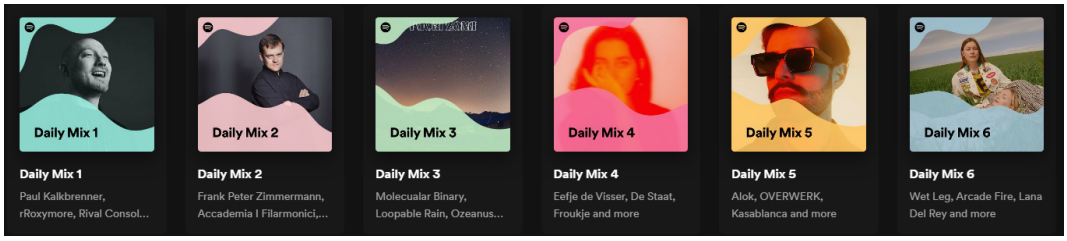


Figure 7.1: The icons for Spotify's automatically generated mixes contain little meaningful visual representation of the music in the playlist.

Bibliography

Pablo Alonso-Jiménez, Xavier Serra, and Dmitry Bogdanov. Music representation learning based on editorial metadata from discogs. 2022.

Ehsan Amid and Manfred K Warmuth. Trimap: Large-scale dimensionality reduction using triplets. *arXiv preprint arXiv:1910.00204*, 2019.

Ashton Anderson, Lucas Maystre, Ian Anderson, Rishabh Mehrotra, and Mounia Lalmas. Algorithmic effects on the diversity of consumption on spotify. In *Proceedings of The Web Conference 2020*, pages 2155–2165, 2020.

Mihael Ankerst. Visual data mining with pixel-oriented visualization techniques. In *Proceedings of the ACM SIGKDD Workshop on Visual Data Mining*, page 23, 2001.

Lilac Atassi. Hinged t-sne for musical interfaces. In *International Conference on New Interfaces for Musical Expression*.

Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23, 1981.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Ethan Benjamin and Jaan Altosaar. Musicmapper: interactive 2d representations of music samples for in-browser remixing and exploration. In *NIME*, pages 325–326, 2015.

Jacques Bertin. *Semiology of graphics*. University of Wisconsin press, 1983.

Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. 2011.

Frédéric Blanchard, Michel Herbin, and Laurent Lucas. A new pixel-oriented visualization technique through color image. *Information Visualization*, 4(4):257–265, 2005.

Simon J Blanchard and Ishani Banerji. Evidence-based recommendations for designing free-sorting experiments. *Behavior research methods*, 48:1318–1336, 2016.

Rita Borgo, Johannes Kehrer, David HS Chung, Eamonn Maguire, Robert S Laramee, Helwig Hauser, Matthew Ward, and Min Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (State of the Art Reports)*, pages 39–63, 2013.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.

Richard Brath. Multiple shape attributes in information visualization: Guidance from prior art and experiments. In *2010 14th International Conference Information Visualisation*, pages 433–438. IEEE, 2010.

Matthew Brehmer, Robert Kosara, and Carmen Hull. Generative design inspiration for glyphs with diatoms. *arXiv preprint arXiv:2107.09015*, 2021.

John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

Lukas Brunner, Jochen Flachhuber, Lukas Kurzmann, and Tobias Striemitzer. Survey of color for information visualisation guidelines for choosing the right colors in information visualisation. 2019.

Li Cai. Latent variable modeling. *Shanghai archives of psychiatry*, 24(2):118, 2012.

Pritish Chandna, Marius Miron, Jordi Janer, and Emilia Gómez. Monoaural audio source separation using deep convolutional neural networks. In *International conference on latent variable analysis and signal separation*, pages 258–266. Springer, 2017.

Jian Chen, Petra Isenberg, Robert S Laramee, Tobias Isenberg, Michael Sedlmair, Torsten Mōller, and Han-Wei Shen. Not as easy as you think-experiences and lessons learnt from creating a visualization image typology. 2022.

Ke Chen, Beici Liang, Xiaoshuan Ma, and Minwei Gu. Learning audio embeddings with user listening data for content-based music recommendation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3015–3019. IEEE, 2021.

Min Chen, Luciano Floridi, and Rita Borgo. What is visualization really for? In *The Philosophy of Information Quality*, pages 75–93. Springer, 2014.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Ya-Xi Chen and René Klüber. Thumbnaildj: Visual thumbnails of music content. In *ISMIR*, pages 565–570, 2010.

Shenghui Cheng, Wei Xu, Wen Zhong, and Klaus Mueller. A data-driven approach for mapping multivariate data to color. *arXiv preprint arXiv:1608.05772*, 2016.

Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American statistical Association*, 68(342):361–368, 1973.

Asher Tobin Chodos et al. What does music mean to spotify? an essay on musical significance in the era of digital curation. *INSAM Journal of Contemporary Music, Art and Technology*, 1(2):36–64, 2019.

Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *arXiv preprint arXiv:1703.09179*, 2017.

Paul W Cleary and Mark L Sawley. Dem modelling of industrial granular flows: 3d case studies and the effect of particle shape on hopper discharge. *Applied Mathematical Modelling*, 26(2):89–111, 2002.

Douglas W Cunningham and Christian Wallraven. *Experimental design: From user studies to psychophysics*. CRC Press, 2011.

Willem C De Leeuw and Jarke J van Wijk. A probe for local flow field visualization. In *Proceedings Visualization'93*, pages 39–45. IEEE, 1993.

MC Ferreira De Oliveira and Haim Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE transactions on visualization and computer graphics*, 9(3):378–394, 2003.

Cyril De Runz, Eric Desjardin, and Michel Herbin. Unsupervised visual data mining using self-organizing maps and a data-driven color mapping. In *2012 16th International Conference on Information Visualisation*, pages 241–245. IEEE, 2012.

Navneet Dhand and Mehar Singh. Sample size calculator for comparing paired differences. URL `https://statulator.com/SampleSize/ss2PM.html#`.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Justin Donaldson. Music recommendation mapping and interface based on structural network entropy. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 811–817. IEEE, 2007.

Simon Durand and Daniel Stoller. 3.3 what is the best task to learn a generic music audio representation? *Deep Learning and Knowledge Integration for Music Audio Analysis*, page 112, 2022.

Stephen HC DuToit, A Gert W Steyn, and Rolf H Stumpf. *Graphical exploratory data analysis*. Springer Science & Business Media, 2012.

Bianchi Dy, Nazim Ibrahim, Ate Poorthuis, and Sam Joyce. Improving visualization design for effective multi-objective decision making. *IEEE Transactions on Visualization and Computer Graphics*, 28(10):3405–3416, 2021.

Howard Egeth and Robert Pachella. Multidimensional stimulus identification. *Perception & Psychophysics*, 5(6):341–346, 1969.

James Elder and Steven Zucker. The effect of contour closure on the rapid discrimination of two-dimensional shapes. *Vision research*, 33(7):981–991, 1993.

Daniel PW Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The quest for ground truth in musical artist similarity. 2002.

Charles W Eriksen and Harold W Hake. Absolute judgments as a function of stimulus range and number of stimulus and response categories. *Journal of Experimental Psychology*, 49(5):323, 1955.

David Feng, Yueh Lee, Lester Kwock, and Russell M Taylor. Evaluation of glyph-based multivariate scalar volume visualization techniques. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization*, pages 61–68, 2009.

Rodolfo Figueroa. Spotify 1.2m+ songs, Dec 2020. URL `https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs`.

Camilla Forsell, Stefan Seipel, and Mats Lind. Simple 3d glyphs for spatial multivariate data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 119–124. IEEE, 2005.

Steven L Franconeri, Lace M Padilla, Priti Shah, Jeffrey M Zacks, and Jessica Hullman. The science of visual data communication: What works. *Psychological Science in the public interest*, 22(3):110–161, 2021.

Michael Friendly. Statistical graphics for multivariate data. *SAS SUGI*, 16:1157–1162, 1991.

Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, Fabian Fischer, and Enrico Bertini. The influence of contour on similarity perception of star glyphs. *IEEE transactions on visualization and computer graphics*, 20(12):2251–2260, 2014.

Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, and Daniel Keim. A systematic review of experimental studies on data glyphs. *IEEE transactions on visualization and computer graphics*, 23(7):1863–1879, 2016.

Johannes Hermann Fuchs. *Glyph design for temporal and multi-dimensional data: Design considerations and evaluation*. PhD thesis, 2015.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Tim Gerrits, Christian Rössl, and Holger Theisel. Glyphs for general second-order 2d and 3d tensors. *IEEE transactions on visualization and computer graphics*, 23(1):980–989, 2016.

Michael Gleicher. Considerations for visualizing comparison. *IEEE transactions on visualization and computer graphics*, 24(1):413–423, 2017.

Joshua S Gulmatico, Julie Ann B Susa, Mon Arjay F Malbog, Aimee Acoba, Marte D Nipas, and Jennalyn N Mindoro. Spotipred: A machine learning approach prediction of spotify music popularity by audio features. In *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, pages 1–5. IEEE, 2022.

Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, volume 10, pages 339–344. Citeseer, 2010.

Yoonchang Han, Jaehun Kim, and Kyogu Lee. Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):208–221, 2016.

Christopher G Healey and James T Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE transactions on visualization and computer graphics*, 5(2):145–167, 1999.

Andre Holzapfel, Bob Sturm, and Mark Coeckelbergh. Ethical dimensions of music information retrieval technology. *Transactions of the International Society for Music Information Retrieval*, 1(1):44–55, 2018.

Christine Hosey, Lara Vujović, Brian St. Thomas, Jean Garcia-Gathright, and Jennifer Thom. Just give me what i want: How people use and evaluate music search. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

Yihan Hou, Haotian Zhu, Hai-Ning Liang, and Lingyun Yu. A study of the effect of star glyph parameters on value estimation and comparison. *Journal of Visualization*, pages 1–15, 2022.

Ruizhen Hu, Bin Chen, Juzhan Xu, Oliver Van Kaick, Oliver Deussen, and Hui Huang. Shape-driven coordinate ordering for star glyph sets via reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(6):3034–3047, 2021.

James Jackson, Panagiotis D Ritsos, and Jonathan C Roberts. Creating small unit based glyph visualisations. In *Posters presented at the IEEE Conference on Visualization (IEEE VIS 2018), Berlin, Germany (Oct. 2018)*, volume 2, 2018.

Il-Young Jeong and Kyogu Lee. Learning temporal features using a deep neural network and its application to music genre classification. In *Ismir*, pages 434–440, 2016.

Samruddhi Y Kahu, Rajesh B Raut, and Kishor M Bhurchandi. Review and evaluation of color spaces for image/video compression. *Color Research & Application*, 44(1):8–33, 2019.

Dietrich Kammer, Mandy Keck, Thomas Gründer, Alexander Maasch, Thomas Thom, Martin Kleinsteuber, and Rainer Groh. Glyphboard: Visual exploration of high-dimensional data combining glyphs with dimensionality reduction. *IEEE transactions on visualization and computer graphics*, 26(4):1661–1671, 2020.

SP. Kane. Post. clincalc: https://clincalc.com/stats/power.aspx., 2018.

Mandy Keck and Lars Engeln. Sparkle glyphs: A glyph design for the analysis of temporal multivariate audio features. In *Proceedings of the 2022 International Conference on Advanced Visual Interfaces*, pages 1–3, 2022.

Mandy Keck, Dietrich Kammer, Thomas Gründer, Thomas Thom, Martin Kleinsteuber, Alexander Maasch, and Rainer Groh. Towards glyph-based visualizations for big data clustering. In *Proceedings of the 10th international symposium on visual information communication and interaction*, pages 129–136, 2017.

Muhammed Khawatmi, Yoann Steux, Saddam Zourob, and Heba Z Sailem. Shapography: A user-friendly web application for creating bespoke and intuitive visualisations of biomaging data using a glyph-oriented approach. *Frontiers in Bioinformatics*, page 64, 2022.

Richard Khulusi, Jakob Kusnick, Christofer Meinecke, Christina Gillmann, Josef Focht, and Stefan Jänicke. A survey on visualizations for musical data. In *Computer Graphics Forum*, volume 39, pages 82–110. Wiley Online Library, 2020.

Hyun-Ju Kim, Min-Joon Yoo, Ji-Yong Kwon, and In-Kwon Lee. Generating affective music icons in the emotion plane. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3389–3394. 2009.

Jaehun Kim, Minz Won, Xavier Serra, and Cynthia CS Liem. Transfer learning of artist group factors to musical genre classification. In *Companion Proceedings of the The Web Conference 2018*, pages 1929–1934, 2018.

Jaehun Kim, Julián Urbano, Cynthia Liem, and Alan Hanjalic. One deep music representation to rule them all? a comparative analysis of different representation learning strategies. *Neural Computing and Applications*, 32(4):1067–1093, 2020.

Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *IEEE transactions on visualization and computer graphics*, 12(5):1329–1336, 2006.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Klippel, Frank Hardisty, Rui Li, and Chris Weaver. Colour-enhanced star plot glyphs: Can salient shape characteristics be overcome? *Cartographica: The International Journal for Geographic Information and Geovisualization*, 44(3):217–231, 2009a.

Alexander Klippel, Frank Hardisty, and Chris Weaver. Star plots: How shape characteristics influence classification tasks. *Cartography and Geographic Information Science*, 36(2): 149–163, 2009b.

Peter Knees and Kristina Andersen. Searching for audio by sketching mental images of sound: A brave new idea for audio retrieval in creative music production. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 95–102, 2016.

Peter Knees and Markus Schedl. *Music similarity and retrieval: an introduction to audio- and web-based strategies*, volume 9. Springer, 2016.

Peter Knees, Markus Schedl, and Masataka Goto. Intelligent user interfaces for music discovery. *Transactions of the International Society for Music Information Retrieval*, 3 (1), 2020.

Philipp Kolhoff, Jacqueline Preuß, and Jörn Loviscach. Content-based icons for music files. *Computers & Graphics*, 32(5):550–560, 2008.

Andreas Kopf and Manfred Claassen. Latent representation learning in biology and translational medicine. *Patterns*, 2(3):100198, 2021.

Gunther Kress and Theo Van Leeuwen. *Reading images: The grammar of visual design*. Routledge, 2020.

Kaori Kusama and Takayuki Itoh. Muscat: a music browser featuring abstract pictures and zooming user interface. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 1222–1228, 2011.

Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Junaid Qadir, and Björn W Schuller. Deep representation learning in speech processing: Challenges, recent advances, and future trends. *arXiv preprint arXiv:2001.00378*, 2020.

Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, pages 387–392, 2009.

Harin Lee, Frank Hoeger, Marc Schoenwiesner, Minsu Park, and Nori Jacoby. Cross-cultural mood perception in pop songs and its alignment with mood detection algorithms. *arXiv preprint arXiv:2108.00768*, 2021.

Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789*, 2017.

Jongpil Lee, Jiyoung Park, and Juhan Nam. Representation learning of music using artist, album, and track information. *arXiv preprint arXiv:1906.11783*, 2019.

John P Lewis, Ruth Rosenholtz, Nickson Fong, and Ulrich Neumann. Visualids: automatic distinctive icons for desktop interfaces. *ACM Transactions on Graphics (TOG)*, 23(3): 416–423, 2004.

Hugo Lima, Carlos Santos, and Bianchi Meiguins. Visualizing the semantics of music. In *2019 23rd International Conference Information Visualisation (IV)*, pages 352–357. IEEE, 2019.

Matteo Lionello, Luca Pietrogrande, Hendrik Purwins, and Mohamed Abou-Zleikha. Interactive exploration of musical space with parametric t-sne. In *15th Sound and Music Computing Conference (SMC 2018)*, pages 200–208. Sound and Music Computing Network, 2018.

Xuejiao Luo, Leonardo Scandolo, and Elmar Eisemann. Texture browser: Feature-based texture exploration. In *Computer Graphics Forum*, volume 40, pages 99–109. Wiley Online Library, 2021.

Alan M MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 2004.

Wakako Machida and Takayuki Itoh. Lyricon: A visual music selection interface featuring multiple icons. In *2011 15th International Conference on Information Visualisation*, pages 145–150. IEEE, 2011.

Eamonn Maguire, Philippe Rocca-Serra, Susanna-Assunta Sansone, Jim Davies, and Min Chen. Taxonomy-based glyph design—with a case study on visualizing workflows of biological experiments. *IEEE Transactions on Visualization and Computer Graphics*, 18 (12):2603–2612, 2012.

Sergio Oramas Martín. *Knowledge Extraction and Representation Learning for Music Recommendation and Classification*. PhD thesis, Ph. D. thesis, Universitat Pompeu Fabra.[Cited on page 139.], 2017.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Alessandro B Melchiorre, Navid Rekabsaz, Emilia Parada-Cabaleiro, Stefan Brandl, Oleg Lesota, and Markus Schedl. Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing & Management*, 58(5):102666, 2021.

G Metternicht and J Stott. Trivariate spectral encoding: a prototype system for automated selection of colours for soil maps based on soil textural composition. In *Proceedings of the 21st International Cartographic Conference, Durban, CD*, 2003.

Matthias Miller, Xuan Zhang, Johannes Fuchs, and Michael Blumenschein. Evaluating ordering strategies of star glyph axes. In *2019 IEEE Visualization Conference (VIS)*, pages 91–95. IEEE, 2019.

Jill Morton. *A guide to color symbolism*, volume 28. Colorcom, 1997.

Chris Muelder, Thomas Provan, and Kwan-Liu Ma. Content based graph visualization of audio data for music library navigation. In *2010 IEEE International Symposium on Multimedia*, pages 129–136. IEEE, 2010.

Meinard Müller, Rachel Bittner, and Juhan Nam. Deep learning and knowledge integration for music audio analysis (dagstuhl seminar 22082). In *Dagstuhl Reports*, volume 12. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

Tamara Munzner. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6):921–928, 2009.

Tamara Munzner. *Visualization analysis and design*. CRC press, 2014.

Rutger Nijkamp. Prediction of product success: explaining song popularity by audio features from spotify data. B.S. thesis, University of Twente, 2018.

Christine Nothelfer, Michael Gleicher, and Steven Franconeri. Redundant coding can speed up segmentation in multiclass displays. *IEEE Visualization Poster Proceedings*, 2016.

Christine Nothelfer, Michael Gleicher, and Steven Franconeri. Redundant encoding strengthens segmentation and grouping in visual displays of data. *Journal of Experimental Psychology: Human Perception and Performance*, 43(9):1667, 2017.

Mizuho Oda and Takayuki Itoh. Mist: a music icon selection technique using neural network. *NICOGRAPH International*, 2007, 2007.

Tomasz Opach, Stanislav Popelka, Jitka Dolezalova, and Jan Ketil Rød. Star and polyline glyphs in a grid plot and on a map display: which perform better? *Cartography and Geographic Information Science*, 45(5):400–419, 2018.

Stephen E Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999.

Jiyoung Park, Jongpil Lee, Jangyeon Park, Jung-Woo Ha, and Juhan Nam. Representation learning of music using artist labels. *arXiv preprint arXiv:1710.06648*, 2017.

Ashis Pati and Alexander Lerch. Is disentanglement enough? on latent representations for controllable music generation. *arXiv preprint arXiv:2108.01450*, 2021.

Fernando Vieira Paulovich, Danilo Medeiros Eler, Jorge Poco, Charl P Botha, Rosane Minghim, and Luis Gustavo Nonato. Piece wise laplacian-based projection for interactive data exploration and organization. In *Computer Graphics Forum*, volume 30, pages 1091–1100. Wiley Online Library, 2011.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901.

Wei Peng, Matthew O Ward, and Elke A Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *IEEE Symposium on Information Visualization*, pages 89–96. IEEE, 2004.

Binh Pham. Spline-based color sequences for univariate, bivariate and trivariate mapping. In *Proceedings of the First IEEE Conference on Visualization: Visualization90*, pages 202–208. IEEE, 1990.

Pavlin Poličar. opentsne parameter guide, 2020. URL `https://opentsne.readthedocs.io/en/latest/parameters.html`.

Frank J Post, Theo van Walsum, Frits H Post, and Deborah Silver. Iconic techniques for feature visualization. In *Proceedings Visualization'95*, pages 288–295. IEEE, 1995.

Dr.nbsp;Pamela Reynolds. Principles of data visualization, Apr 2021. URL `https://ucdavisdatalab.github.io/workshop_data_viz_principles/principles-of-visual-perception.html`.

Theresa-Marie Rhyne. Applying color theory to digital media and visualization. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1264–1267, 2017.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning*, pages 4364–4373. PMLR, 2018.

Timo Ropinski, Michael Specht, Jennis Meyer-Spradow, Klaus H Hinrichs, and Bernhard Preim. Surface glyphs for visualizing multimodal volume data. In *VMV*, volume 1, pages 3–12, 2007.

Antonia Saravanou, Federico Tomasi, Rishabh Mehrotra, and Mounia Lalmas. Multi-task learning of graph-based inductive representations of music content. In *ISMIR*, pages 602–609, 2021.

Markus Schedl. Intelligent user interfaces for social music discovery and exploration of large-scale music repositories. In *Proceedings of the 2017 ACM Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces*, pages 7–11, 2017.

Markus Schedl, Arthur Flexer, and Julián Urbano. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539, 2013.

Markus Schedl, Emilia Gómez, Julián Urbano, et al. Music information retrieval: Recent developments and applications. *Foundations and Trends® in Information Retrieval*, 8 (2-3):127–261, 2014.

Thomas Schultz and Gordon Kindlmann. A maximum enhancing higher-order tensor glyph. In *Computer Graphics Forum*, volume 29, pages 1143–1152. Wiley Online Library, 2010a.

Thomas Schultz and Gordon L Kindlmann. Superquadric glyphs for symmetric second-order tensors. *IEEE transactions on visualization and computer graphics*, 16(6):1595–1604, 2010b.

Mariangela Sciandra and Irene Carola Spera. A model-based approach to spotify data analysis: a beta glmm. *Journal of Applied Statistics*, 49(1):214–229, 2022.

Vidya Setlur, Conrad Albrecht-Buehler, Amy A. Gooch, Sam Rossoff, and Bruce Gooch. Semanticons: Visual metaphors as file icons. In *Computer Graphics Forum*, volume 24, pages 647–656. Blackwell Publishing, Inc Oxford, UK and Boston, USA, 2005.

Dougal Shakespeare, Lorenzo Porcaro, Emilia Gómez, and Carlos Castillo. Exploring artist gender bias in music recommendation. *arXiv preprint arXiv:2009.01715*, 2020.

Christopher D Shaw, David S Ebert, James M Kukla, Amen Zwa, Ian Soboroff, and D Aaron Roberts. Data visualization using automatic perceptually motivated shapes. In *Visual Data Exploration and Analysis V*, volume 3298, pages 208–213. SPIE, 1998.

Jingyi Shen, Runqi Wang, and Han-Wei Shen. Visual exploration of latent space for traditional chinese music. *Visual Informatics*, 4(2):99–108, 2020.

Petter Skidén. Api improvements and u, 2016. URL `https://developer.spotify.com/community/news/2016/03/29/api-improvements-update/`.

Janne Spijkervet and John Ashley Burgoyne. Contrastive learning of musical representations. *arXiv preprint arXiv:2103.09410*, 2021.

David Sprague, Fuqu Wu, and Melanie Tory. Music selection using the partyvote democratic jukebox. In *Proceedings of the working conference on Advanced visual interfaces*, pages 433–436, 2008.

Tadeusz Stach, Carl Gutwin, David Pinelle, and Pourang Irani. Improving recognition and characterization in groupware with rich embodiments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 11–20, 2007.

Maureen Stone. *A field guide to digital color*. CRC Press, 2003.

Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel, Daniel A Keim, and Oliver Deussen. Document cards: A top trumps visualization for documents. *IEEE transactions on visualization and computer graphics*, 15(6):1145–1152, 2009.

Bob L Sturm. The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*, 2013.

Robert E Thayer. *The biopsychology of mood and arousal*. Oxford University Press, 1990.

Edward R Tufte, Nora Hillman Goeler, and Richard Benson. *Envisioning information*, volume 126. Graphics press Cheshire, CT, 1990.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

Tomomi Uota and Takayuki Itoh. Grape: A gradation based portable visual playlist. In *2014 18th International Conference on Information Visualisation*, pages 361–365. IEEE, 2014.

Mark van de Ruit, Markus Billeter, and Elmar Eisemann. An efficient dual-hierarchy t-sne minimization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1): 614–622, 2021.

Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. *Advances in neural information processing systems*, 26, 2013.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *The Journal of Machine Learning Research*, 22(1):9129–9201, 2021.

Matthew O Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

Matthew O Ward. Multivariate data glyphs: Principles and practice. In *Handbook of data visualization*, pages 179–198. Springer, 2008.

Colin Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2019.

Colin Ware, Francesca Samsel, David H Rogers, Paul A Navrátil, and Ayat Mohammed. Designing pairs of colormaps for visualizing bivariate scalar fields. In *EuroVis (Short Papers)*, pages 49–53, 2020.

Chris Weigle and Russel M Taylor. Visualizing intersecting surfaces with nested-surface techniques. In *VIS 05. IEEE Visualization, 2005.*, pages 503–510. IEEE, 2005.

Jeremy M Wolfe. Visual search. 2015.

Minz Won et al. *Representation learning for music classification and retrieval: bridging the gap between natural language and music semantics*. PhD thesis, Universitat Pompeu Fabra, 2022.

Lu Ying, Tan Tangl, Yuzhe Luo, Lvkeshen Shen, Xiao Xie, Lingyun Yu, and Yingcai Wu. Glyphcreator: Towards example-based automatic generation of circular glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):400–410, 2021.

Lu Ying, Xinhuan Shu, Dazhen Deng, Yuchen Yang, Tan Tang, Lingyun Yu, and Yingcai Wu. Metaglyph: Automatic generation of metaphoric glyph-based visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

K Yoshii and M Goto. Visualizing musical pieces in thumbnail images based on acoustic features. In *9th International Conference on Music Information Retrieval (ISMIR)*, pages 211–216, 2007.

Mason Youngblood, Yuto Ozaki, and Patrick E Savage. Cultural evolution and music. 2021.

Wenqi Zhong, Aibing Yu, Xuejiao Liu, Zhenbo Tong, and Hao Zhang. Dem/cfd-dem modelling of non-spherical particulate systems: theoretical developments and applications. *Powder technology*, 302:108–152, 2016.

Appendix A

Terminology

Bias

Bias is a statistical term to describe statistics that fail to offer an accurate portrayal of the population.

Dimensionality Reduction

Dimensionality reduction is a transformation of data from a high-dimensional space into a lower-dimensional space. The objective is for the lower-dimensional representation to maintain as much properties of the original data as possible.

Downstream Task

A downstream task is a task that relies on the output of a preceding task or process.

Features

In computer science and multimedia analysis, music characteristics are modelled as 'features'. In this document, we refer to all information extracted from a raw signal as features. Any feature may serve as input for other (machine learning) algorithms.

International Society for Music Information Retrieval (ISMIR)

The International Society for Music Information Retrieval (ISMIR) serves as an international platform for research related to the organisation of music-related data.

Latent variables

Latent variables are (random) variables which are not directly observed or cannot be measured, sometimes also referred to as hidden variables. Latent variables are often inferred from the observed variables.

Latent variable models

A latent variable model is a statistical model that contains latent variables [Cai, 2012]. Examples of models that explicitly model latent variables are Hidden Markov Models and neural networks. As deep learning models have a lot of unobserved variables, we consider them latent variable models as well.

Music Information Retrieval (MIR)

MIR is the research field concerned with the extraction and inference of meaningful features from music.

Pre-attentive properties

Pre-attentive properties are visual characteristics that are processed without requiring conscious effort, prior to the conscious mind's engagement.

Appendix B

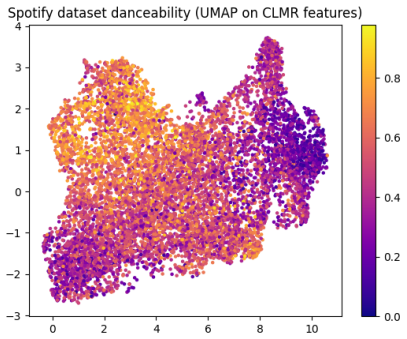
Model selection

Name	**Authors**	**Year**	**Model type**	**Framework**	**Task**
Contrastive learning of musical representations	Spijkervet et al	2021	CNN	Pytorch	classification
Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders	Engel et al	2017	CNN	Magenta	generation
Vector-quantized timbre representation	Bitton et al	2020	VQ-VAE	Pytorch	timbre modeling
Semi-supervised music tagging transformer	Won et al	2021	Transformer & CNN	Pytorch	classification
One deep music representation to rule them all?	Kim et al	2020	CNN	Pytorch	multitask / generalisation
MusiCNN	Pons et al	2019	CNN	Tensorflow	classification
Transfer learning for music classification and regression tasks	Choi et al	2017	CNN	Theano & Keras	classification / regression
Automatic Tagging Using Deep Convolutional Neural Networks	Choi et al	2016	CNN	Tensorflow & Keras	classification
Toward interpretable music tagging with self-attention.	Won et al	2019	Transformer	Pytorch	classification
Jukebox	Dhariwal et al	2020	Transformer	Pytorch	generation

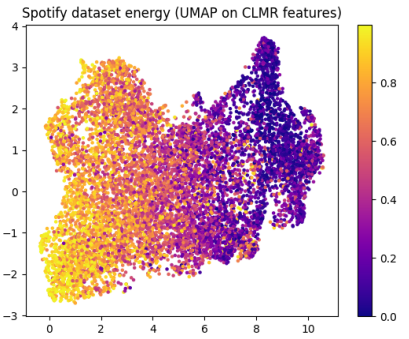
Table B.1: List of suitable models for our visualisation task.

Genre	**Artist**	**Album name**
classic	Bach	A Musical Genius
classic	Vivalidi	The Four Seasons
rap	Eminem	The Eminem Show
rap	Nas	Illmatic
rock/metal	Nirvana	Nevermind
rock/metal	Slipknot	Iowa
techno	Paul Kalkbrenner	Berlin Calling
techno	Vitalic	Rave Age
country	Waylon Jennings	Dreaming My Dreams
country	Willie Nelson	Red Headed Stranger

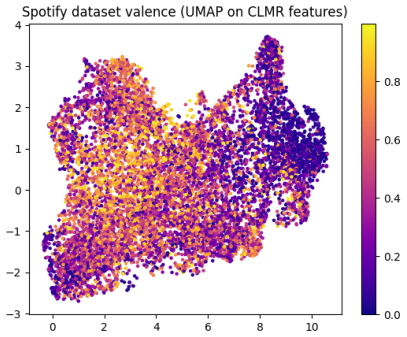
Table B.2: Composition of custom test dataset: 2 iconic albums for each genre.



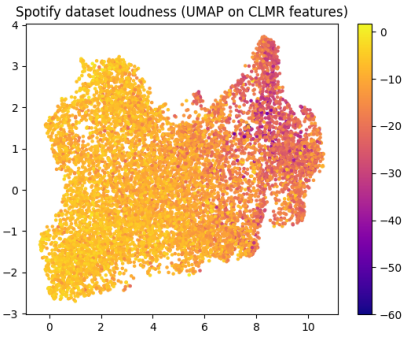
(a) Spotify Danceability Features



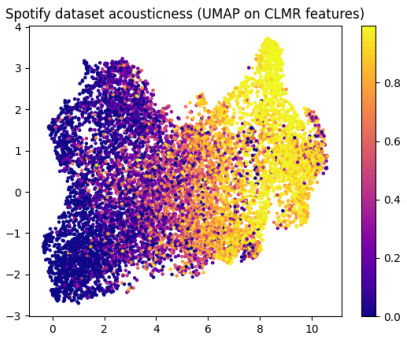
(b) Spotify Energy Features



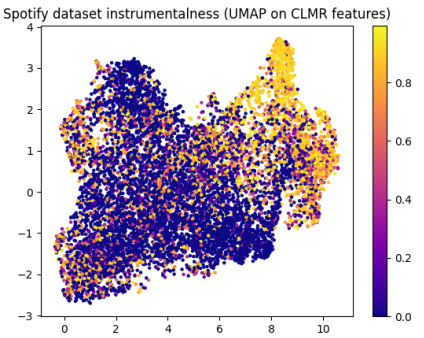
(c) Spotify Valence Features



(d) Spotify Loudness Features

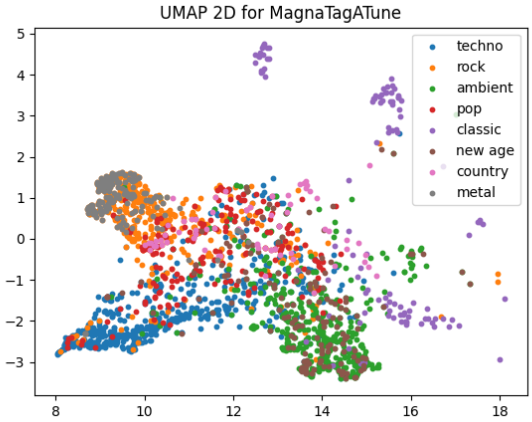


(e) Spotify Acousticness Features

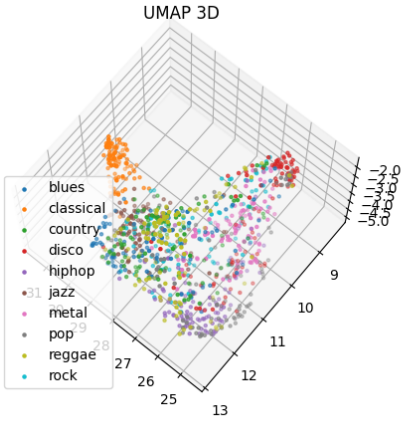


(f) Spotify Instrumentalness Features

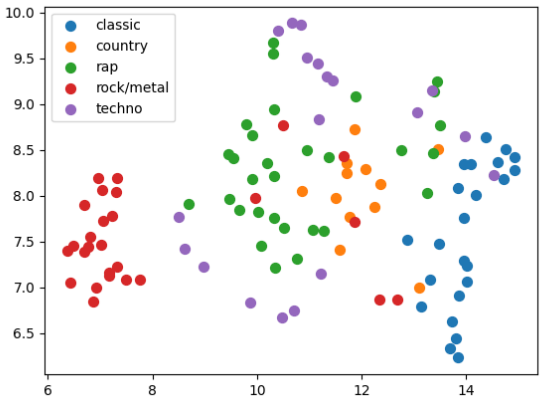
Figure B.2: 2D UMAP embeddings of features extracted from Spijkervet and Burgoyne [2021] for different features of the custom Spotify dataset based on Figueroa [2020].



(a) MagnaTagATune dataset



(b) GTZAN dataset



(c) Custom dataset

Figure B.1: UMAP embeddings of features extracted from Spijkervet and Burgoyne [2021] on three different datasets for genre classification.

Appendix C

Convergence experiments

Algorithm	Average	Median	Std	Min	Max	Calculation time
PCA	0.952172	0.974872	0.065124	0.013444	0.991046	3
t-SNE	0.179058	0.185255	0.083984	-0.161475	0.373460	4523
UMAP	0.961708	0.962517	0.008550	0.889408	0.983136	49
PaCMAP	0.128631	0.163214	0.126748	-0.257000	0.335293	28
TriMap	0.127323	0.172649	0.137623	-0.273397	0.351861	38

Table C.1: Comparison of similarity of vectors after reducing 512 to 8 dimensions for 5 different algorithms. Calculation time in seconds.

Perplexity	Average Similarity	Median Similarity	Std Similarity	Min Similarity	Max Similarity	Calculation time
1000	0.17268	0.18552	0.08899	-0.18056	0.34522	4212
500	0.18579	0.20104	0.11598	-0.24315	0.39561	3571
100	0.17410	0.18850	0.08425	-0.17394	0.33649	3550
50	0.176194	0.18386	0.079181	-0.150540	0.349156	3851
40	0.182017	0.18620	0.087062	-0.153931	0.384679	4423
30	0.183074	0.18705	0.091390	-0.163330	0.392327	4166
20	0.178602	0.18448	0.081814	-0.155643	0.372543	4380
10	0.17536	0.18307	0.08474	-0.16426	0.36785	4443
5	0.167323	0.17122	0.068370	-0.129227	0.344086	5541

Table C.2: Exploration of the perplexity hyperparameter on the t-SNE algorithm. Calculation times in seconds

Nearest Neighbours	Min dist	Average Similarity	Median Similarity	Std Similarity	Min Similarity	Max Similarity	Calculation time
100	0.00	0.961993	0.962819	0.008473	0.889785	0.983121	59
100	0.10	0.962184	0.963055	0.008397	0.890453	0.983125	64
100	0.50	0.962302	0.963131	0.008373	0.890888	0.983186	62
100	0.85	0.962515	0.963342	0.008325	0.891364	0.983268	47
100	0.99	0.962865	0.963728	0.008223	0.892174	0.983290	59
50	0.00	0.961678	0.962511	0.008539	0.889420	0.983155	46
50	0.10	0.962190	0.963047	0.008405	0.890339	0.983156	56
50	0.50	0.961999	0.962823	0.008576	0.888803	0.983173	57
50	0.85	0.962637	0.963491	0.008321	0.891480	0.983249	56
50	0.99	0.963240	0.964120	0.008166	0.892780	0.983313	52
25	0.00	0.965147	0.966255	0.007757	0.897988	0.984012	41
25	0.10	0.965161	0.966269	0.007757	0.897875	0.983983	40
25	0.50	0.965255	0.966262	0.007741	0.897339	0.984319	41
25	0.85	0.965225	0.966170	0.007712	0.897333	0.983554	38
25	0.99	0.964957	0.965895	0.007753	0.896742	0.983313	37
15	0.00	0.961709	0.962487	0.008548	0.889346	0.983107	33
15	0.10	0.961508	0.962280	0.008624	0.888752	0.983118	32
15	0.50	0.961880	0.962681	0.008512	0.889886	0.983169	31
15	0.85	0.962192	0.963015	0.008429	0.890693	0.983203	31
15	0.99	0.962279	0.963096	0.008423	0.890654	0.983204	31
10	0.00	0.963933	0.964950	0.008017	0.895463	0.983441	28
10	0.10	0.963991	0.965017	0.008002	0.895396	0.983417	26
10	0.50	0.964129	0.965124	0.007998	0.895215	0.983231	27
10	0.85	0.964573	0.965505	0.007962	0.895822	0.983648	30
10	0.99	0.965297	0.966130	0.007804	0.895897	0.984625	29
5	0.00	0.963988	0.964989	0.008033	0.895696	0.983799	19
5	0.10	0.963769	0.964781	0.008064	0.894953	0.983400	18
5	0.50	0.963712	0.964685	0.008057	0.894731	0.983294	17
5	0.85	0.963986	0.964932	0.007989	0.895851	0.983520	19
5	0.99	0.964174	0.965121	0.007948	0.896557	0.983700	19

Table C.3: Exploration of the Nearest Neighbour and Minimum distance hyperparameters on the UMAP algorithm.

Appendix D

Glyph Design

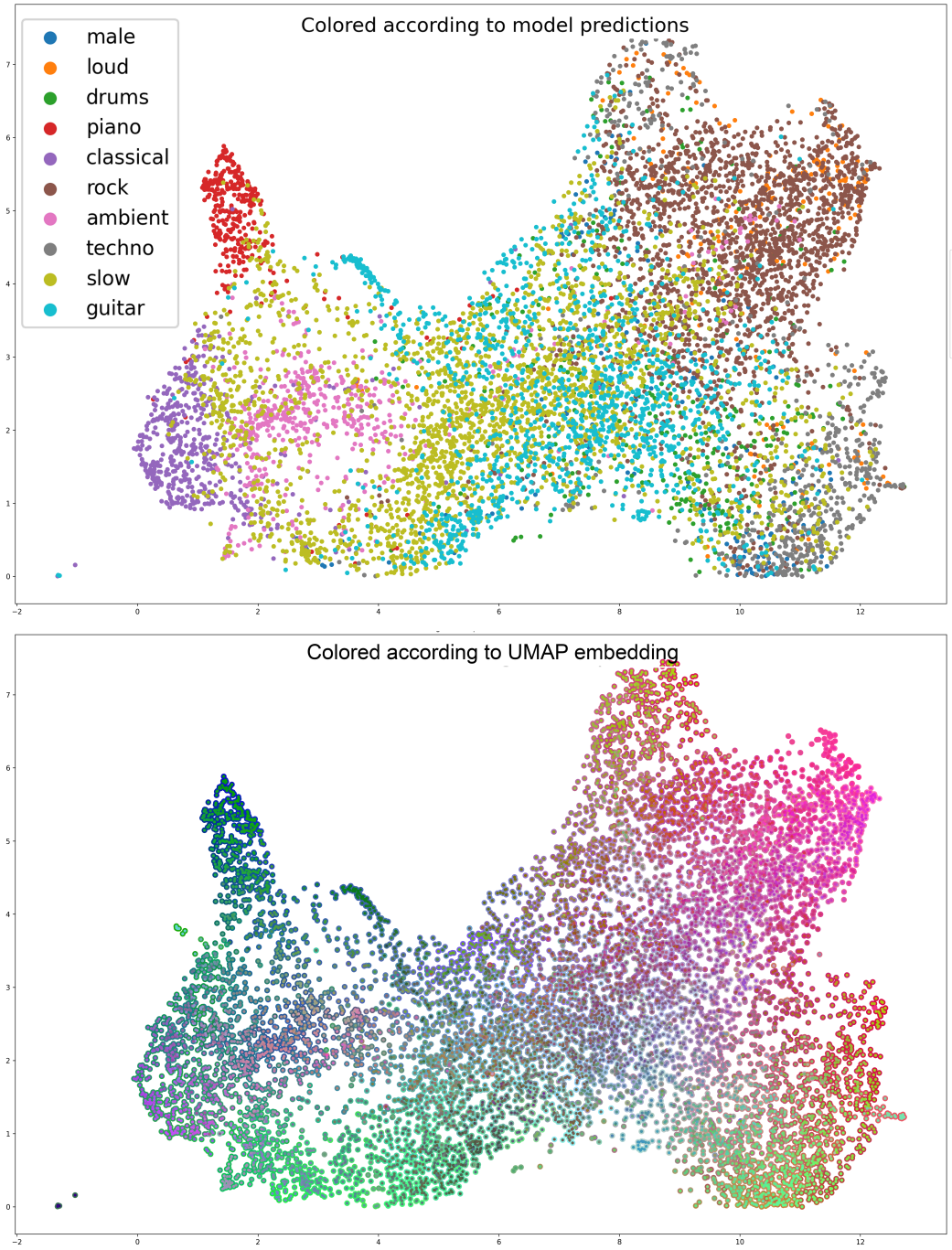
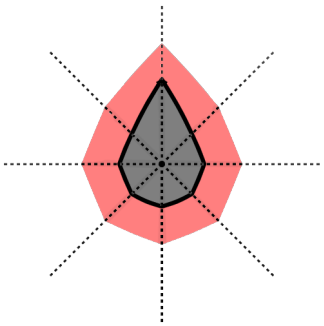
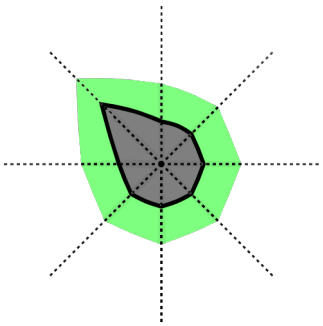


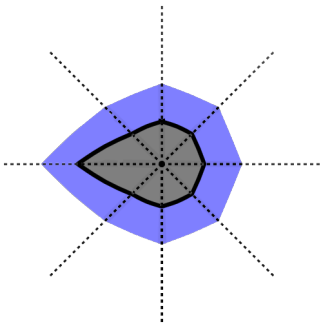
Figure D.1: Comparison of the top 10 most predicted labels vs colours based on UMAP embedding of representations. Dataset: Spotify dataset based on Figueroa [2020].



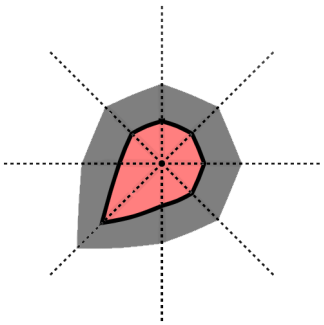
(a) Icon for values: [1.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]



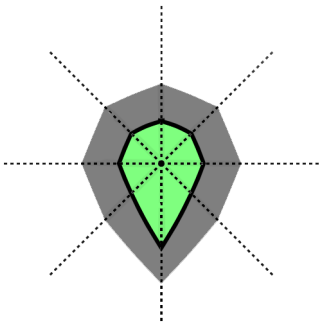
(b) Icon for values: [0.5, 1.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]



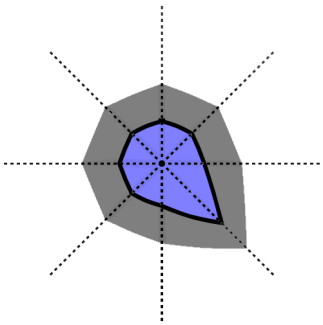
(c) Icon for values: [0.5, 0.5, 0.5, 1.0, 0.5, 0.5, 0.5, 0.5]



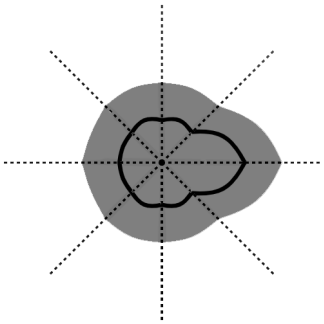
(d) Icon for values: [0.5, 0.5, 0.5, 0.5, 1.0, 0.5, 0.5, 0.5]



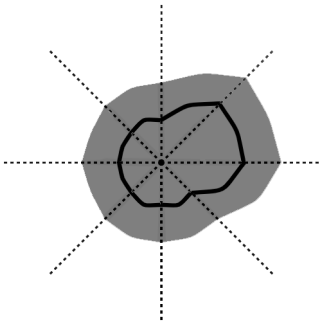
(e) Icon for values: [0.5, 0.5, 0.5, 0.5, 1.0, 0.5, 0.5, 0.5]



(f) Icon for values: [0.5, 0.5, 0.5, 0.5, 0.5, 1.0, 0.5, 0.5]



(g) Icon for values: [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1.0, 0.5]



(h) Icon for values: [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 1.0, 1.0]

Figure D.2: The influence of each parameter with redundant encoding in the glyph design.

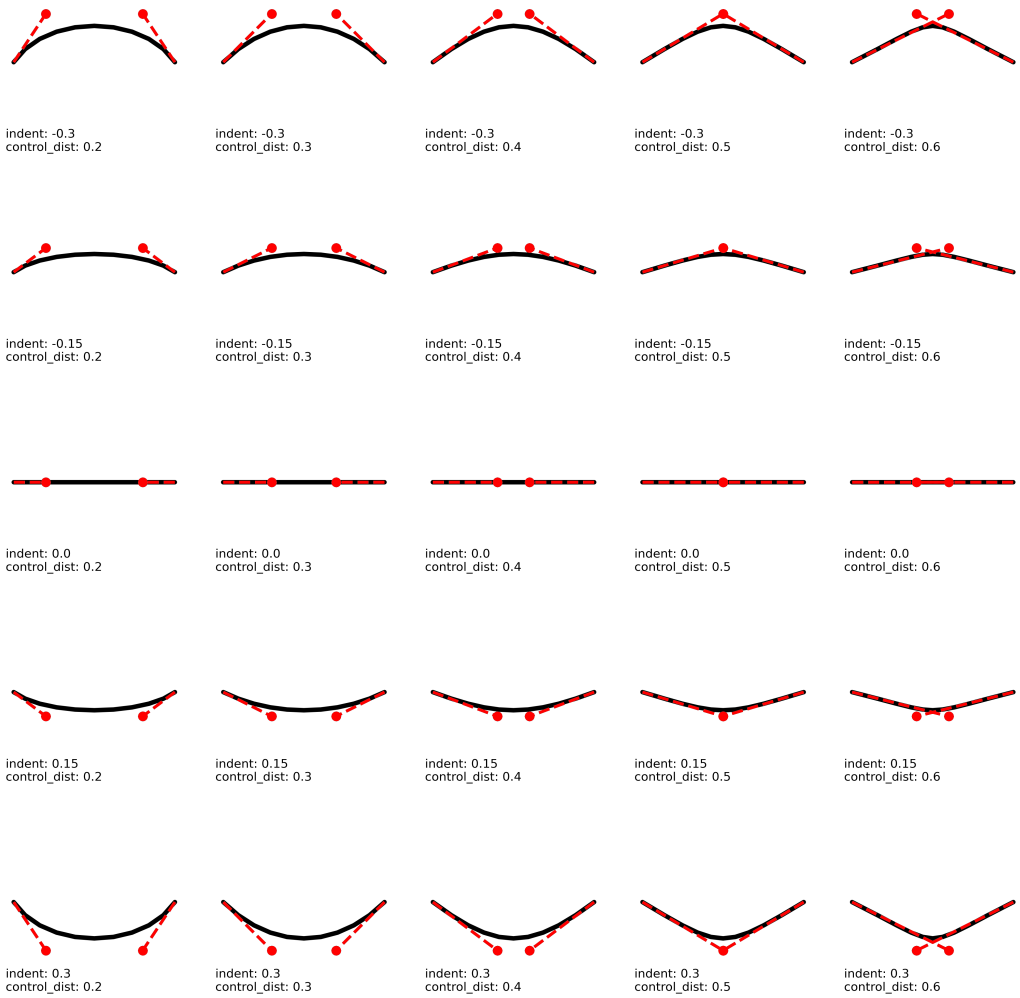


Figure D.3: Effects of parameter settings on curves constructed: on the x axis the control distance is varied, along the y axis the direction and strength.

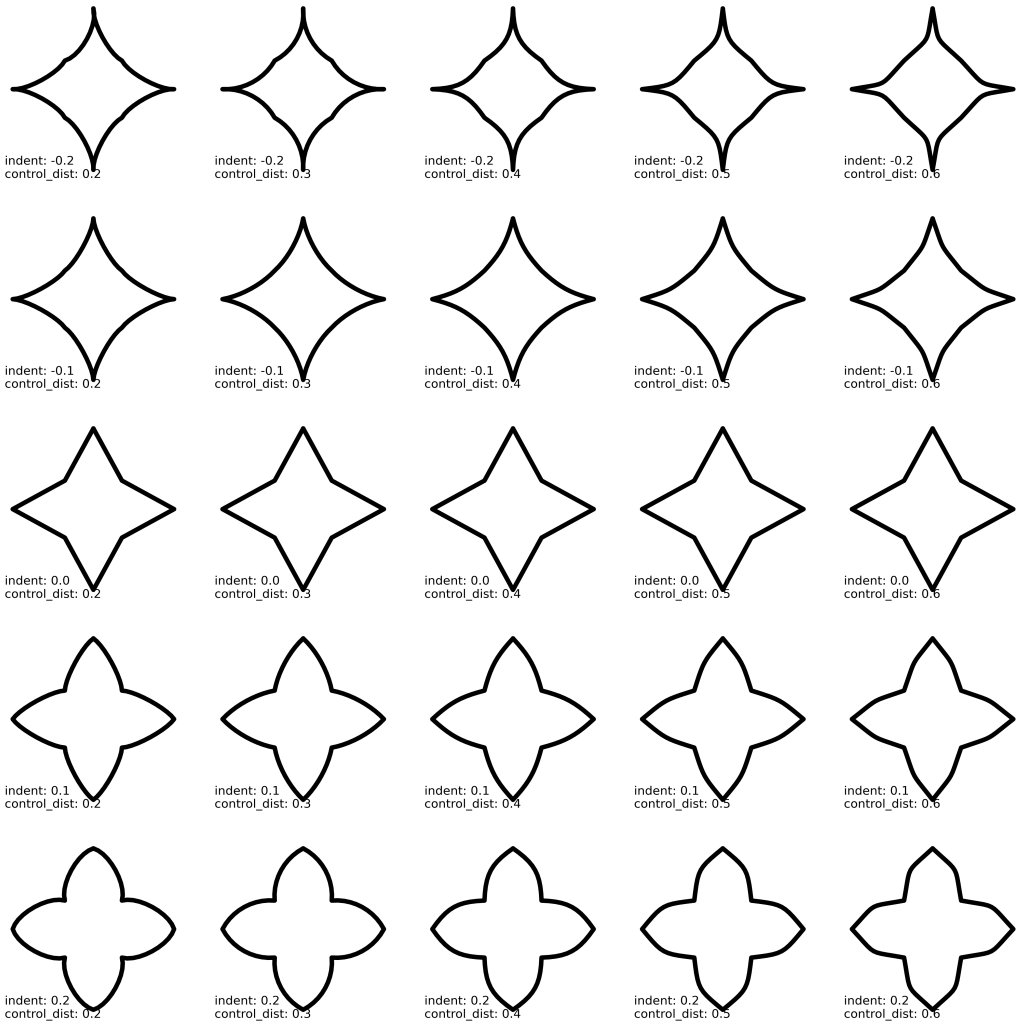


Figure D.4: Influence of curve parameters on an 8 dimensional star shape: on the x axis the control distance is varied, along the y axis the direction and strength.

(a) Icons for test set sorted on 1d UMAP embedding of 512 dimensional features.

(b) Icons for test set sorted on 1d UMAP embedding of the 8d UMAP embedding of the features, on which the icon is also based.

(c) Icons for test set sorted on a 1d PCA of the 8d UMAP embedding of the features on which the icon is also based

Figure D.5: Experiments with sorting methods.

Appendix E

User Tests

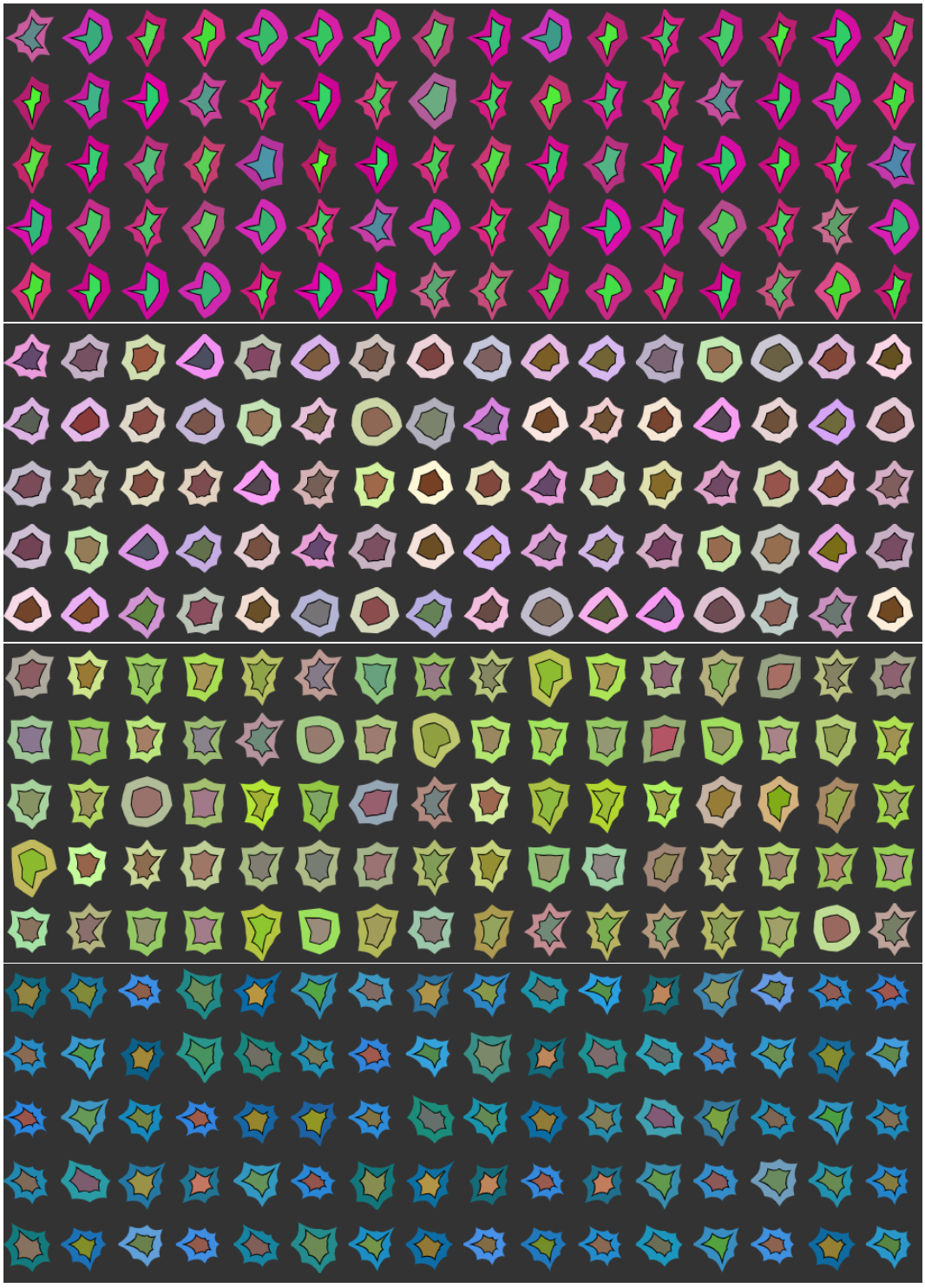


Figure E.1: Random samples from 4 of the 10 clusters that were obtained with k-means clustering on the 10.000 song Spotify dataset.

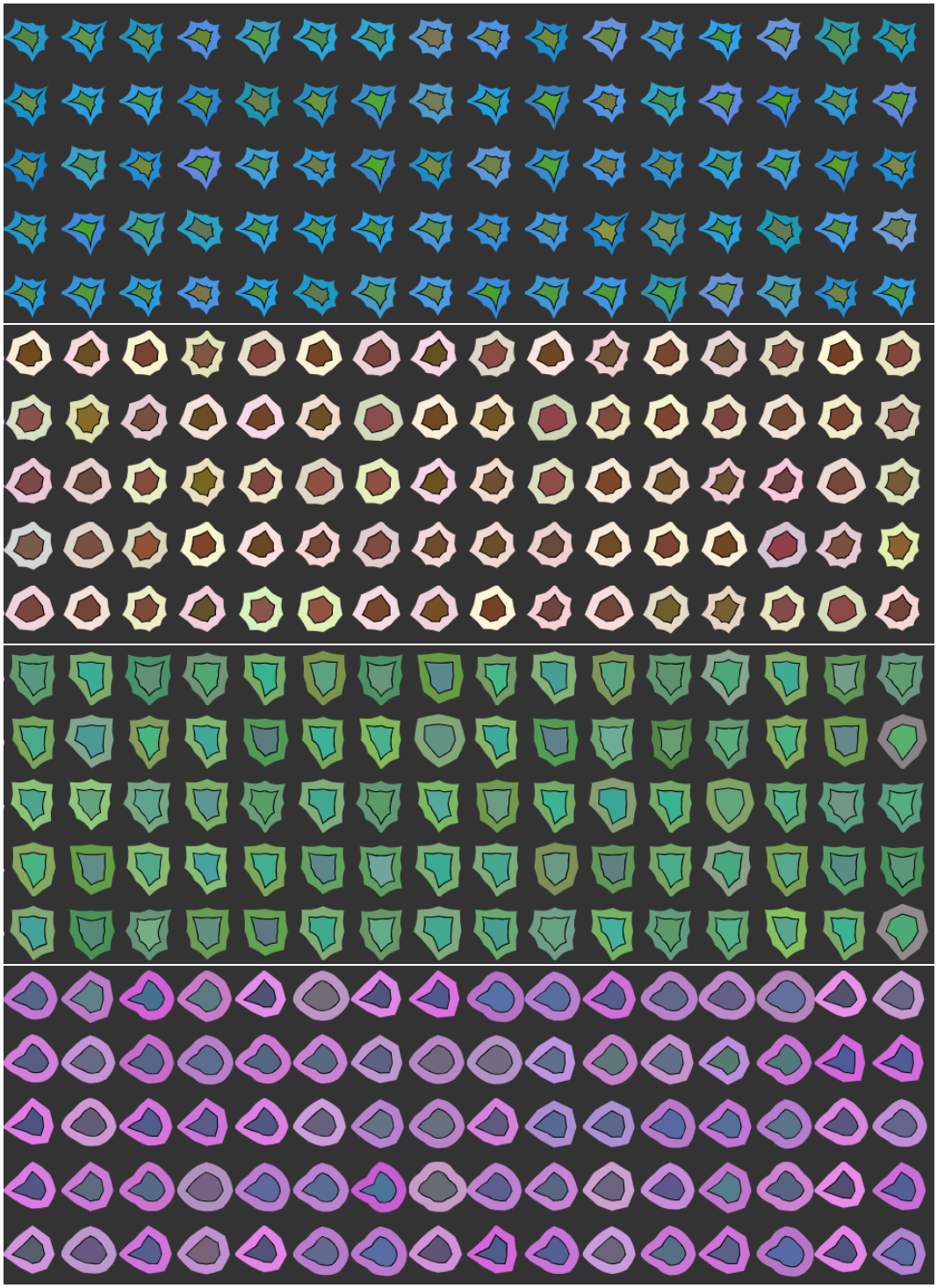


Figure E.2: Random samples from 4 of the 40 clusters that were obtained with k-means clustering on the 10.000 song Spotify dataset.

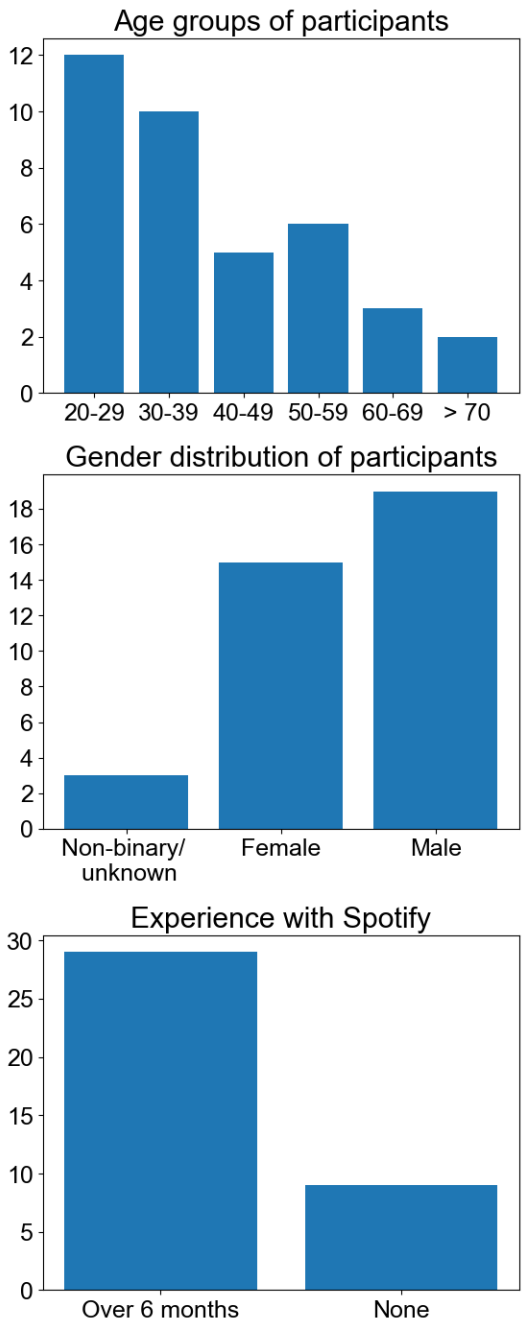


Figure E.3: Demographic information on the participants of the user study.

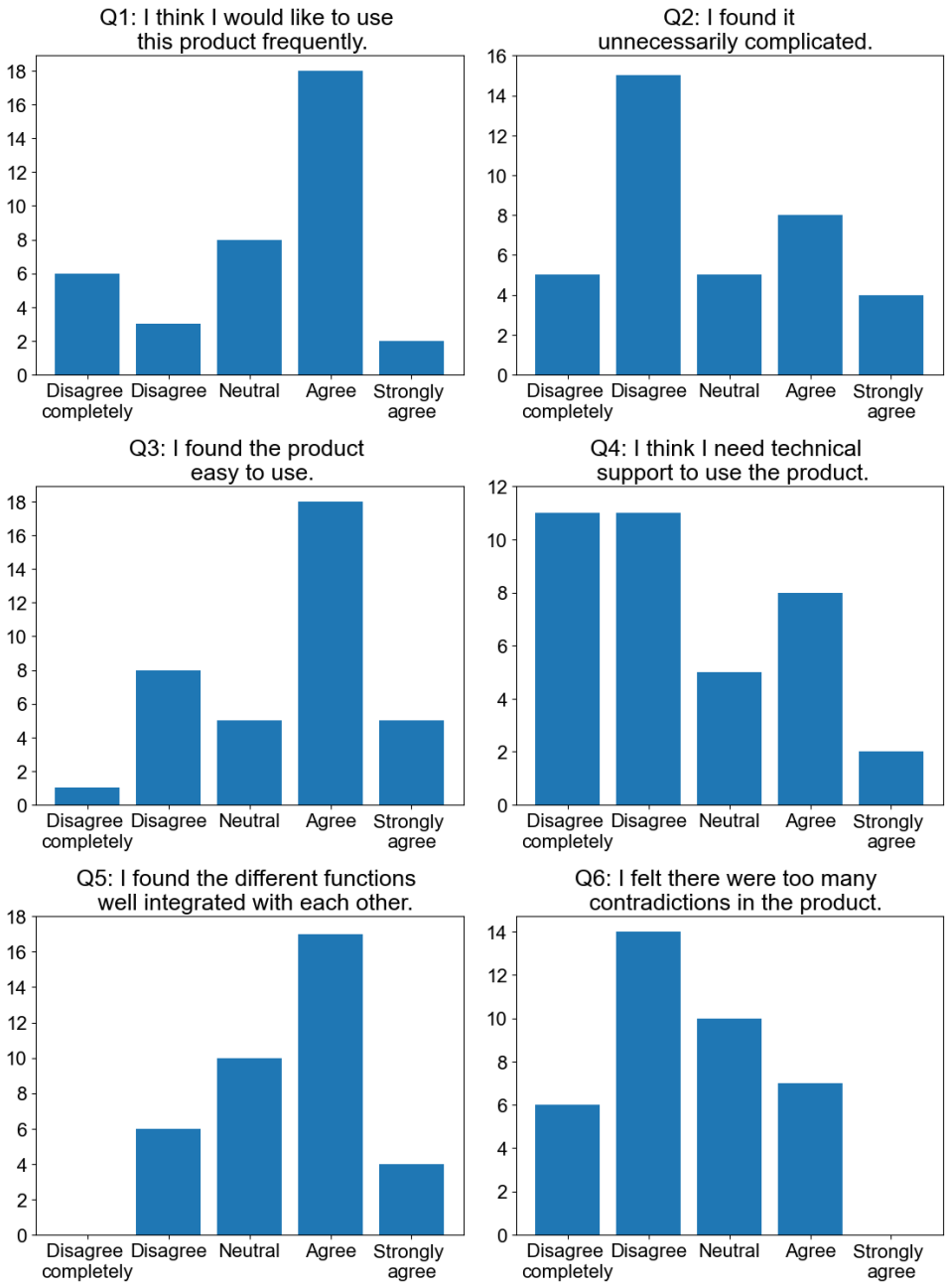


Figure E.4: Answers given to questions 1 to 6 of the SUS

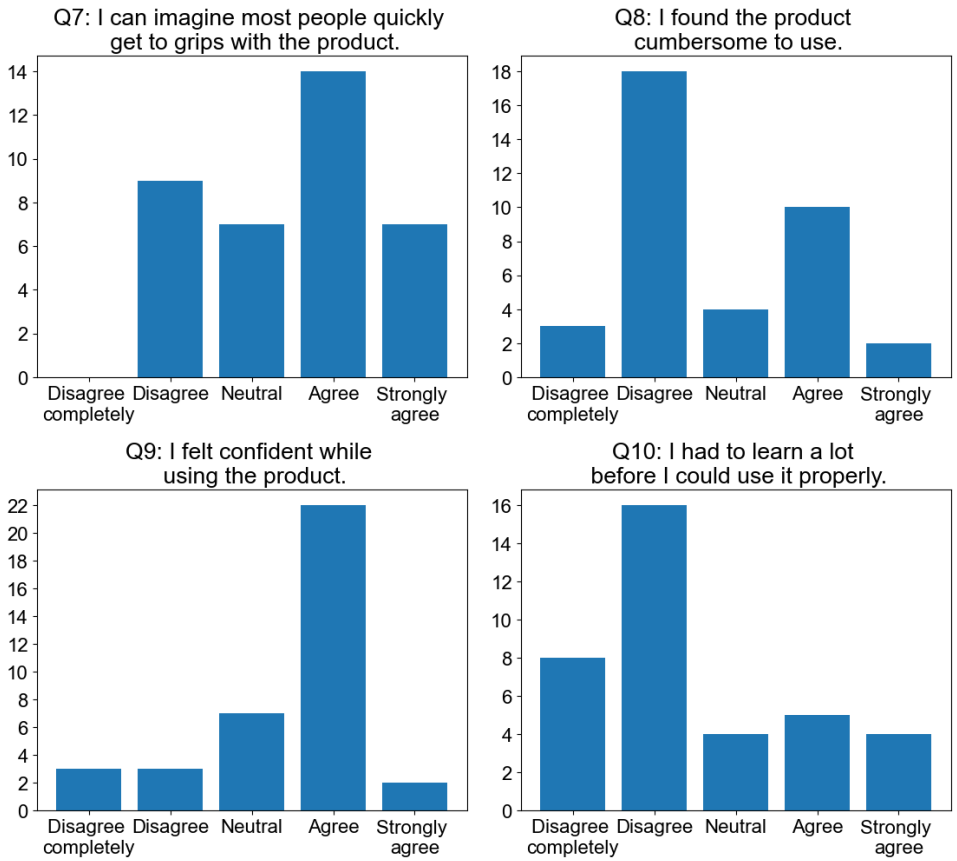


Figure E.5: Answers given to questions 7 to 10 of the SUS